

- Stagiaire : **LACOMBE Alexia**  
a8lacomb@enib.fr

**Mise en place d'un environnement de réalité virtuelle et/ou  
de réalité augmentée permettant la mise en oeuvre des  
protocoles de traçabilité et d'hygiène**

- Maître de stage : **DANIEL Tanguy**  
Directeur informatique  
tanguy.daniel@emanrisk.fr

  
**e-ManRisk**  
ZI DE KERSCAO  
29480 LE RELECQ-KERHUON  
SIRET 820 843 001 00026 APE 7490B  
contact@emanrisk.fr

## Remerciements

Je souhaite dans un premier temps, remercier la société e-ManRisk et notamment son directeur, CONSTANS David, pour m'avoir accueillie pour ce stage .

Je remercie également DANIEL Tanguy, directeur informatique, de m'avoir intégrée à son service, d'avoir été mon maître de stage et pour l'expérience et les conseils apportés tout le long de celui-ci.

Je tiens à remercier aussi QUERREC Ronan, directeur du CERV, pour son aide apportée concernant la scénarisation et Mascaret et d'être mon tuteur de stage.

Je remercie l'ensemble des équipes d'e-ManRisk pour l'accueil chaleureux et les bons moments et conseils au quotidien.

Je remercie enfin ma famille et mes amis pour l'aide à la rédaction et la relecture de ce rapport.

## Résumé

Dans le cadre de mon S8 à l'ENIB (Ecole National d'Ingénieur de Brest), j'ai effectué mon stage assistant ingénieur du 14 février au 1er juillet 2022 au sein du département informatique de l'entreprise e-ManRisk.

e-ManRisk est une filiale du groupe E.Leclerc qui acte dans le domaine de la gestion des risques et du développement logiciel.

L'objectif de ce stage était la mise en place d'un environnement de réalité virtuelle et/ou de réalité augmentée permettant la mise en œuvre des protocoles de traçabilité et d'hygiène.

J'ai ainsi travaillé autour de 3 parties:

- La modélisation d'un environnement de réalité virtuelle sur Unity, notamment sur la partie animation des objets mais aussi interactions entre les objets et le participant.
- L'élaboration et la mise en place d'un scénario pour cette formation axé sur la procédure de cuisson, refroidissement rapide et stockage d'un aliment à l'aide de Mascaret, un système développé par le CERV.
- L'intégration du logiciel développé par la société e-ManRisk au sein de l'environnement 3D et la possibilité d'interagir avec celui-ci à l'aide de la librairie node: Puppeteer.

Souhaitant me spécialiser dans le domaine de l'informatique et plus précisément de la réalité virtuelle ou du jeu vidéo, j'ai choisi ce stage car il s'inscrit parfaitement dans mon projet professionnel.

Finalement, ce stage correspondait totalement à mes attentes et m'a conforté dans mes choix d'orientation pour l'année prochaine: réaliser un double diplôme "Maîtrise en informatique option Jeux Vidéo" à l'UQAC (Université Québécoise à Chicoutimi) et dans mon envie de travailler dans les jeux vidéos sérieux.

## Abstract

As part of my semester eight at ENIB (national engineering school of Brest), I did my assistant engineer internship from February 14 to July 1 2022 in the IT department of the company e-ManRisk.

e-ManRisk is a subsidiary of the E.Leclerc group which acts in the field of risk management and software development.

The objective of this internship was to set up a virtual reality and/or augmented reality environment allowing the implementation of traceability and hygiene protocols.

My internship was structured around 3 parts:

- The modeling of a virtual reality environment on Unity, in particular on the animation part of objects but also interactions between the objects and the participant.
- The development and implementation of a training scenario focused on the cooking procedure, rapid cooling and storage of food using Mascaret, a system developed by CERV.
- The integration of the software developed by the company e-ManRisk within the 3D environment and the possibility of interacting with it using the Node library: Puppeteer.

Wishing to specialize in the field of computer science and more specifically virtual reality or video games, I chose this internship because it fits perfectly into my professional project.

Finally, this internship totally fulfilled my expectations and confirmed my choice of orientation for next year: to complete a double degree "Masters in Computer Science option Video Games" at UQAC (Quebec University in Chicoutimi) and in my interest in working on serious video games.

## Sommaire

TABLEAU DES FIGURES	6
GLOSSAIRE	7
1. INTRODUCTION	8
2. PRÉSENTATION DE L'ENTREPRISE	9
2.1. E-MANRISK, SOCIÉTÉ DE SERVICE	9
2.2. SON ORGANISATION	9
2.3. UN UNIQUE CLIENT : LE MOUVEMENT E.LECLERC	10
2.4. SON ACTIVITÉ : SEQUOIA	10
3. CONTEXTE	11
4. PLANIFICATION	12
5. DÉVELOPPEMENT DU SUJET DE STAGE	13
5.1. CHOIX DU TYPE DE FORMATION	13
5.2. CHOIX D'UN OUTIL POUR LA SCÉNARISATION	14
5.3. MISE EN PLACE DE MASCARET	17
5.4. LA GESTION DU TEMPS	24
5.5. LA GESTION DE LA CONTAMINATION ALIMENTAIRE	25
5.6. LA MISE EN PLACE DU LOGICIEL D'E-MANRISK DANS L'ENVIRONNEMENT VIRTUEL	26
6. PERSPECTIVES FUTURES DU PROJET	36
6.1. LA FORMATION EN RÉALITÉ VIRTUELLE CHEZ E-MANRISK	36
6.2. LES TESTS FONCTIONNELS FRONT-END	36
7. BILAN PERSONNEL	37
7.1. BILAN DES COMPÉTENCES	37
7.2. ORIENTATION FUTUR	37
8. CONCLUSION	38
9. BIBLIOGRAPHIE	39
10. ANNEXES	40
10.1. SCÉNARIO DU PROJET	40
10.2. LES OUTILS UTILISÉS	46

## Table des figures

FIGURE 1 : ORGANISATION DE L'ENTREPRISE	9
FIGURE 2 : DIAGRAMME DE GANTT	12
FIGURE 3 : EXPLICATION D'UNE PRISE DE TEMPÉRATURE DANS UNE FORMATION INITIALE	13
FIGURE 4 : EXPLICATION D'UNE PRISE DE TEMPÉRATURE DANS UN COMPLÉMENT DE FORMATION	14
FIGURE 5 : EVALUATION D'UNE PRISE DE TEMPÉRATURE	14
FIGURE 6 : DESCRIPTION DU SCÉNARIO DANS UN FICHIER XML	15
FIGURE 7 : EXEMPLE D'UN BPMN D'UTILISATION D'UNE SONDE DE THERMOMÈTRE	16
FIGURE 8 : SALLE PRINCIPALE D'UN LABORATOIRE	17
FIGURE 9 : ECHELLE ET ZONE DE LAVAGE DES MAINS D'UN LABORATOIRE	17
FIGURE 10 : ZONE DE STOCKAGE D'UN LABORATOIRE	18
FIGURE 11 : SALLE PRINCIPALE D'UN LABORATOIRE SOUS UNITY	18
FIGURE 12 : ZONE DE STOCKAGE D'UN LABORATOIRE SOUS UNITY	18
FIGURE 13 : EXEMPLE D'UN DIAGRAMME DE CLASSE	19
FIGURE 14 : EXEMPLE D'UN DIAGRAMME D'ACTIVITÉ : STOCKAGE EN CHAMBRE FROIDE	20
FIGURE 15 : MODÉLISATION D'UN CHOIX DANS UN DIAGRAMME D'ACTIVITÉ	21
FIGURE 16 : CRÉATION D'ENTITÉ	21
FIGURE 17 : AFFECTATION D'ENTITÉ	22
FIGURE 18 : FONCTION EXECUTE() DE L'OPÉRATION "OUVRIRFOUR"	22
FIGURE 19 : INDICATION POUR L'OPÉRATION "OUVRIRFOUR"	23
FIGURE 20 : LES HORLOGES DANS L'ENVIRONNEMENT VIRTUEL	24
FIGURE 21 : ECRAN NOIR POUR PASSER LE TEMPS	25
FIGURE 22 : ALIMENT NON CONTAMINÉ/CONTAMINÉ	25
FIGURE 23 : ARCHITECTURE SERVEUR-CLIENT	27
FIGURE 24 : ARCHITECTURE SERVEUR-CLIENT DE NOTRE SITUATION	27
FIGURE 25 : FORMAT D'UNE REQUÊTE HTTP	28
FIGURE 26 : ROUTE DU SERVEUR	29
FIGURE 27 : ECRAN D'ACCUEIL DE L'APPLICATION D'E-MANRISK	29
FIGURE 28 : EXEMPLE DE LISTE DÉROULANTE DU SITE	30
FIGURE 29 : EXEMPLE DE CODE D'UNE ROUTE	30
FIGURE 30 : EXEMPLE D'ENVOIE D'UNE REQUÊTE AVEC UNITY	31
FIGURE 31 : TABLETTE DANS LE LABORATOIRE	32
FIGURE 32 : TABLETTE INTERACTIVE AVEC UN POINTEUR	33
FIGURE 33 : RÉCUPÉRATION DES CAPTURES D'ÉCRAN	33
FIGURE 34 : RÉCUPÉRATION DU CHOIX DU CLAVIER	34
FIGURE 35 : AFFICHAGE DU CLAVIER	35

## Glossaire

**API (Application Programming Interface)** : est une interface logicielle qui permet de «connecter» un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

**BPMN (Business Process Model and Notation)** : est une norme pour modéliser des processus métier.

**C#** : est un langage de programmation orientée objet.

**Camunda** : est une plate-forme open source d'automatisation des flux de travail et des décisions.

**HTTP (Hypertext Transfer Protocol)** : est un protocole de communications réseaux serveur-client.

**Mascaret (Multi Agent System for Collaborative and Adaptative Realistic Environment for Training)** : est un système développé par le CERV (Centre Européen de Réalité Virtuelle) pour la scénarisation et la gestion de formation en Réalité Virtuelle.

**Modelio** : est un programme open-source pour la réalisation de modèles UML.

**Parseur** : est un programme informatique d'analyse syntaxique permettant d'extraire des éléments d'un fichier.

**Puppeteer** : est une librairie Node qui fournit une API pour contrôler Chrome.

**Requête** : est, en informatique, une demande vers une base de données ou un serveur.

**UML (Unified Modeling Language)** : est un langage de modélisation graphique à base de pictogrammes.

**Unity** : est un logiciel permettant de créer des jeux multiplateformes développé par Unity Technologies.

**XMI (XML Metadata Interchange)** : est un standard pour l'échange d'informations de métadonnées UML basé sur XML.

**XML (eXtensible Markup Language)** : est un métalangage informatique de balisage générique.

## 1. Introduction

Dans le cadre de mon S8 à l'ENIB (Ecole National d'Ingénieur de Brest), j'ai effectué mon stage assistant ingénieur du 14 février au 1er juillet 2022 au sein du département informatique de l'entreprise e-ManRisk.

e-ManRisk est une filiale du groupe E.Leclerc qui acte dans le domaine de la gestion des risques et du développement logiciel.

L'objectif de ce stage était la mise en place d'un environnement de réalité virtuelle et/ou de réalité augmentée permettant la mise en œuvre des protocoles de traçabilité et d'hygiène. Souhaitant me spécialiser dans le domaine de l'informatique et plus précisément de la réalité virtuelle ou du jeu vidéo, j'ai choisi ce stage car il s'inscrit parfaitement dans mon projet professionnel.

Étant toujours en stage au moment de la rédaction de mon rapport, il portera principalement sur mes premiers mois et certains éléments seront amenés à évoluer d'ici la fin de mon stage.

Ce rapport comprendra un glossaire contenant les termes importants, une présentation de l'entreprise e-ManRisk, j'expliquerai ensuite le contexte dans lequel prend place ce stage ainsi que sa planification. Je développerai après le sujet du stage, mes recherches et mes missions ainsi que les perspectives futures de ce projet. Je conclurais sur le bilan personnel. Vous trouverez également en fin de rapport ma bibliographie ainsi que des annexes.

## 2. Présentation de l'entreprise

### 2.1. e-ManRisk, société de service

e-ManRisk, sous sa forme juridique SAS (Société par Actions Simplifiées), a été créée en juin 2016 au sein du Mouvement E. Leclerc et regroupe une équipe expérimentée de plus de 15 ans, dans le domaine de la gestion des risques et du développement logiciel. L'entreprise propose un service complet de management des risques en entreprise, et agit à tous les niveaux d'expertise pour les sociétés du Mouvement E. Leclerc.

### 2.2. Son organisation

Elle comprend 3 services, qui travaillent ensemble :

- **Le pôle informatique** comprenant 5 programmeurs et 1 responsable informatique. Ils ont pour objectifs d'améliorer en continu l'outil de gestion des risques, de résoudre les potentiels soucis, d'ajouter de nouvelles fonctionnalités (modules)...
- **Le pôle risk managers** comprenant 3 personnes. Chacun spécialisé dans un domaine : ressources humaines, qualité-environnement, santé et prévention. Ils ont pour objectif de répondre aux demandes clients, d'élaborer des cahiers des charges à l'intention du service informatique, mais également de réaliser pour les clients des formations à l'utilisation de l'outil, de les sensibiliser aux risques ainsi que le suivi constant.
- **Le directeur et la comptabilité.**

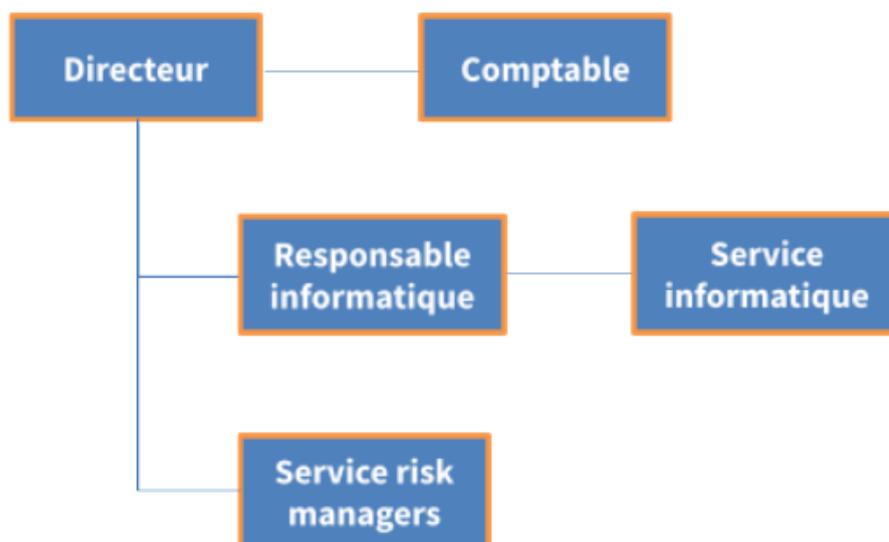


Figure 1 : Organisation de l'entreprise

### 2.3. Un unique client : Le Mouvement E. Leclerc

Créée au sein du mouvement E.Leclerc, la société e-ManRisk conçoit et réalise ses outils logiciels, dont la marque principale est Sequoia, exclusivement pour lui.

Pour mieux comprendre l'implantation de l'outil dans les magasins et les centrales régionales, il faut comprendre le mode de fonctionnement du Mouvement E.Leclerc. Contrairement à d'autres grands groupes comme Carrefour ou encore Auchan, les magasins E.Leclerc sont indépendants, c'est-à-dire qu'ils sont possédés et dirigés par des PDG (Président Directeur Général) qui sont appelés des adhérents.

### 2.4. Son activité : Sequoia

Le projet « Séquoia » est né en 2007 au sein de la Scarmor (région Bretagne), son lancement s'est fait en 2008.

Aujourd'hui, Séquoia est un logiciel de gestion des risques dans cinq domaines :

- **Administratif** : Structures, Annuaire des prestataires, Gestion des contrats, Gestion du personnel, Gestion des aptitudes, Gestion des formations, Entretien professionnels, Gestion des certifications, Actualités,
- **Qualité** : Gestion des retraits, Certificats biologiques, Étalonnage du matériel de mesure, Auto-contrôles, Produits en souffrance (centrale régionale),
- **Sécurité** : Plan de prévention, Contrôle de sécurité, Accueil Sécurité,
- **Environnement** : Registre déchets,
- **Maintenance** : Gestion des installations, Maintenance du matériel.

### 3. Contexte

Mon stage se place dans la continuité d'un précédent stage assistant ingénieur réalisé par LE GALL Marine de février à juillet 2021 dans l'entreprise e-ManRisk. La société souhaite à travers ces stages étudier la possibilité de créer des formations en réalité virtuelle dans le domaine de la gestion des risques au sein du mouvement E. Leclerc qui seraient plus ludiques, interactives et efficaces pour le personnel.

Au cours de son stage, Marine s'est principalement occupée de créer une scène 3D sous Unity représentant un laboratoire dans un magasin, avec l'aide d'un groupe d'étudiants de 5ème année de l'ENIB qui suivait le module PRI. Elle a également réfléchi à la manière de créer un scénario de formation sous Unity, à l'aide d'un exemple conçu par un membre du service métier. Ce scénario idéal se trouve en annexe. Elle a donc intégré une première version linéaire de ce scénario à la réalité virtuelle ainsi que commencé à mettre en place la librairie puppeteer pour utiliser le logiciel d'e-ManRisk dans la formation.

Mon but sera donc d'approfondir la réalisation de ce scénario ainsi que de permettre la simulation du logiciel créé par l'entreprise et qui est utilisé dans les laboratoires des magasins au sein de l'environnement 3D. On souhaite également pouvoir étudier ou non la réussite de l'apprentissage lorsque la personne utilisant cette formation arrive à la fin.

## 4. Planification

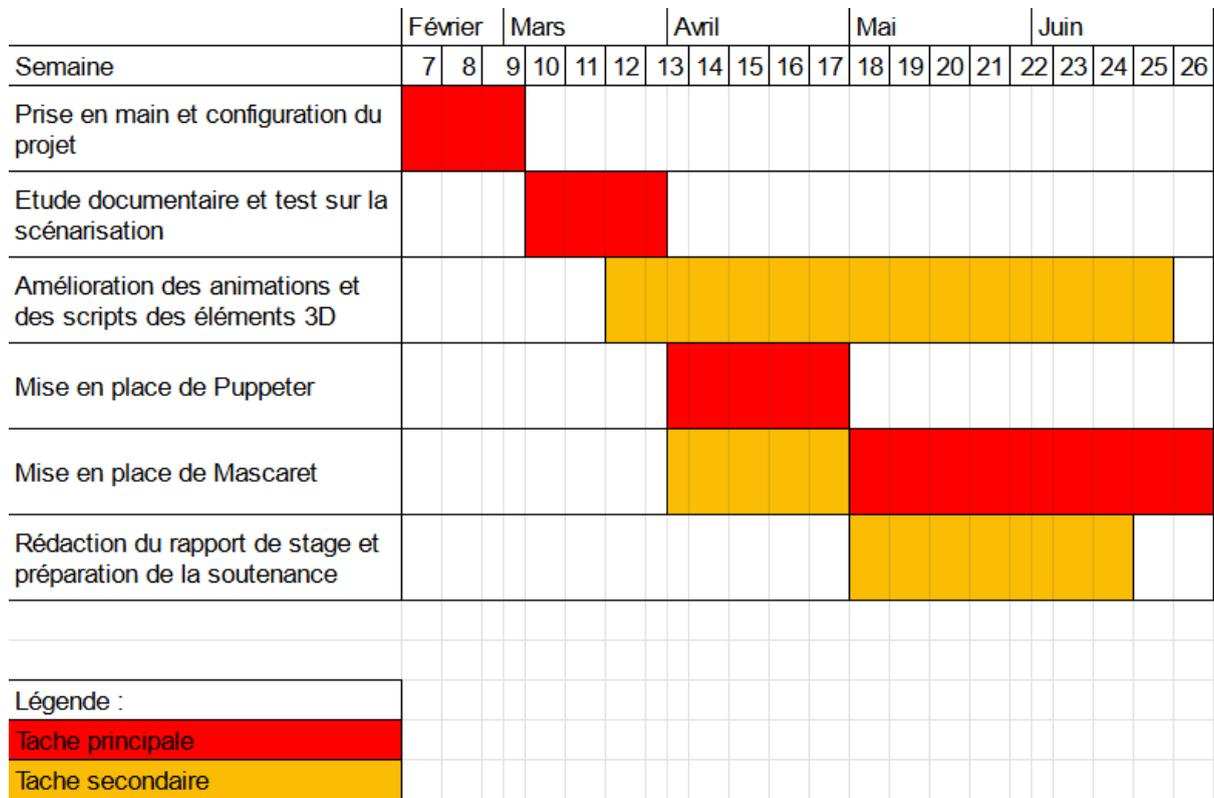


Figure 2 : Diagramme de Gantt

Mon stage a commencé par une prise en main et une configuration du projet existant ainsi que du logiciel Unity . J'ai ensuite pu faire des recherches sur les différentes méthodes permettant de réaliser des scénarios pour une formation en réalité virtuelle.

Ces étapes passées, j'ai pu me concentrer sur la mise en place de *Puppeteer* pour simuler l'application d'e-ManRisk au sein de l'environnement ainsi qu'en parallèle sur l'amélioration du laboratoire virtuel et la mise en place de *Mascaret* pour la scénarisation, qui deviendra ensuite ma tâche principale pour la fin de mon stage.

## 5. Développement du sujet de stage

### 5.1. Choix du type de formation

Avant d'aller plus loin dans la mise en place du scénario, il fallait définir quel type de formation allait-on faire car cela influe sur l'aide apportée à l'apprenant ou sur sa liberté à agir dans l'environnement 3D.

On pouvait distinguer 3 types de formations :

- La formation initiale
- Le complément de formation
- L'évaluation des acquis

Je vais présenter les spécificités de chacune de ses formations.

#### La formation initiale

Une formation initiale prendrait place à l'arrivée d'un employé dans l'entreprise ou dans le service, dans ce cas il ne connaîtrait pas forcément les outils spécifiques, leurs emplacements dans l'environnement ainsi que la manière de les utiliser. Ce type de formation doit en conséquence être très guidée, les objets doivent être mis en évidence, les actions doivent être simples et concises, même pour les plus basiques, afin que l'apprenant puisse se concentrer sur chacune de ses actions, les comprendre et les retenir. On pourrait également limiter la liberté de l'apprenant dans l'espace virtuel en ne l'autorisant qu'à saisir les objets nécessaires à la tâche actuelle.

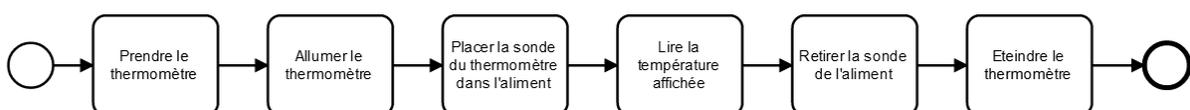


Figure 3 : Explication d'une prise de température dans une formation initiale

#### Le complément de formation

Si la formation sert de complément, on suppose que l'apprenant connaît déjà l'environnement et les tâches à effectuer. Certaines petites instructions deviendront donc implicites, les éléments seront sûrement moins mis en avant. On cherchera surtout à savoir s'il a retenu ses précédentes formations ou à lui en apprendre de nouvelles mais avec du matériel qu'il sait utiliser. On pourra également accorder plus de liberté d'interactions, avec les objets qui l'entourent, à l'apprenant.



Figure 4 : Explication d'une prise de température dans un complément de formation

## L'évaluation des acquis

Si la formation sert plutôt de vérification des acquis, l'apprenant sera libre d'interagir totalement avec l'environnement. Dans ce cas, la personne devrait plutôt réaliser la procédure par lui-même pour qu'ensuite ses actions soient analysées afin de savoir si la procédure est maîtrisée ou s'il faut revenir à des formations initiales ou complémentaires.



Figure 5 : Evaluation d'une prise de température

## Le choix final

Le choix a été fait dans le cadre de ce stage de réaliser une formation initiale : les objets seront indiqués et les actions seront simples afin que la personne puisse assimiler de manière plus efficace les procédures auxquelles on souhaite le former.

### 5.2. Choix d'un outil pour la scénarisation

Le précédent scénario avait été implémenté à l'aide d'un fichier XML qui était parsé, analysé, les actions se déroulaient donc dans l'ordre et les conditions de passage d'une étape à la suivante étaient assez simples : contact entre différents objets 3D par exemple.

Ma principale mission étant d'améliorer ce scénario, il fallait choisir la manière dont il serait implémenté, après quelques recherches 3 solutions se sont présentées :

- Améliorer le *parseur* XML existant afin de pouvoir avoir des actions en parallèles et des conditions plus réalistes
- Utiliser un modèle BPMN (Business Process Model and Notation)
- Utiliser Mascaret

Je vais détailler ces 3 méthodes ainsi que les points positifs et négatifs de chacune afin de conclure sur la méthode utilisée.

## Parseur XML

L'utilisation d'un parseur XML pour l'implémentation du scénario contient 2 étapes caractéristiques : il faut d'une part décrire le scénario dans un fichier XML, choisir quelles informations sont nécessaires pour décrire une étape. Voici un exemple :

```
<Etape id="10">
  <Instruction>Friction en rotation en mouvement de va et vient et les doigts
  | | | | | joints de la main droite dans la paume gauche et vice versa </Instruction>
  <Objet></Objet>
  <Ressources>ldm6</Ressources>
  <Arret>
    <Temps>10</Temps>
    <Evenement></Evenement>
  </Arret>
</Etape>
<Etape id="11">
  <Instruction>Rincer les mains sous l'eau</Instruction>
  <Objet>Sink</Objet>
  <Ressources></Ressources>
  <Arret>
    <Temps>0</Temps>
    <Evenement>Sink/Water</Evenement>
  </Arret>
</Etape>
```

Figure 6 : Description du scénario dans un fichier XML

Chaque étape est ici décrit par :

- un id qui sera unique à chaque étape
- une instruction qui indique que doit faire la personne
- un objet ou plusieurs objets qui doivent être utilisés pour réaliser l'étape et qui seront donc mis en évidence
- une ressource qui peut être une image pour montrer ce qui doit être fait par exemple
- une condition d'arrêt qui est soit un temps ou un événement
  - un temps définirait une durée avant que l'on ne passe automatique à la prochaine étape
  - un événement serait l'accomplissement de l'action de l'étape

La seconde étape serait donc de coder le parseur qui récupérerait ces informations et qui les utiliserait pour avancer dans le scénario à travers le temps et en fonction des actions de la personne formée, mettra en évidence les objets ou les ressources et affichera les instructions.

Cette solution offre une liberté infinie car on peut choisir comment décrire les étapes et comment les réaliser et les enchaîner à l'aide du parseur mais elle est en contrepartie plus longue à mettre en place, ce serait un travail plus lourd et fastidieux, qui réinventerait des concepts sûrement déjà existants. De plus, elle n'est pas visuelle, il est donc plus difficile d'en parler ou de l'expliquer à l'équipe métier.

## BPMN

Il s'agirait dans un premier temps de modéliser notre scénario de formation sous la forme d'un *BPMN*. Ce type de modélisation est très efficace puisqu'elle permet d'avoir des actions qui se réalisent en parallèles ou des décisions à prendre sur le choix de l'étape suivante.

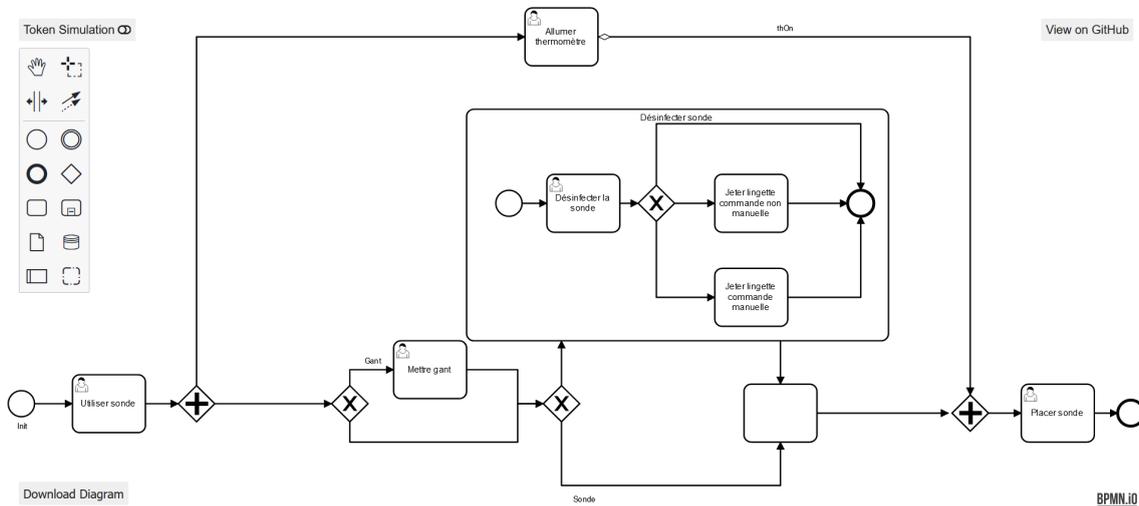


Figure 7 : Exemple de BPMN d'utilisation d'une sonde de thermomètre

Ce modèle pourrait alors être parcouru à l'aide d'un *API* que propose le logiciel Camunda. Certaines actions de l'apprenant enverraient une *requête* qui pourrait faire passer le scénario à la prochaine étape et les instructions seraient ainsi envoyées en réponse par l'*API* à partir du modèle.

Ce type de modélisation est simple et peut être facilement comprise par l'équipe métier ce qui permettrait une meilleure collaboration entre eux et les développeurs lors de la création de scénario. Par contre, cette solution pose des questions de licence dans le cadre d'une industrialisation des formations en réalité virtuelle par e-ManRisk, cette solution ne serait sûrement pas gratuite à long terme.

## Mascaret

Mascaret est un système permettant de mettre en lien des diagrammes *UML* (notamment des diagrammes de classe et d'activité) et un environnement de réalité virtuelle modélisé sur un moteur de jeux 3D comme Unity, afin de réaliser des formations.

Mascaret implémentant le déroulé du scénario avec rigueur, les principales tâches à réaliser seraient de modéliser le scénario avec des diagrammes, de coder les fichiers de configuration permettant de mettre en lien Mascaret et l'environnement de réalité virtuelle puis de coder les actions et les conditions de passage à l'étape suivante.

Cette solution est visuelle comme la précédente car on utilise des diagrammes pour représenter le scénario. De plus, utiliser Mascaret me permettrait d'avoir un cadre plus strict pour élaborer mon scénario sur la durée de mon stage car il se base sur des outils déjà existants tel que la modélisation UML et Mascaret a déjà été utilisé dans de nombreux projets au sein du CERV, cette solution a déjà fait ses preuves, nous avons donc choisi d'utiliser celle ci, que je vais donc détailler dans la suite de mon rapport.

### 5.3. Mise en place de Mascaret

Je vais détailler dans cette partie le processus de mise en place de Mascaret, en abordant principalement les points suivants : la modélisation de l'environnement 3D, la modélisation du scénario en diagramme UML, la mise en relation de l'environnement et des diagrammes et enfin l'implémentation d'une étape.

#### L'environnement 3D

L'environnement dans lequel notre formation prend place est un laboratoire de fabrication au sein d'un magasin du mouvement E. Leclerc. J'ai pu au cours de mon stage me rendre dans ces laboratoires au magasin du Relecq Kerhuon afin d'avoir une meilleure vision des lieux.

Il en existe pour chaque domaine : boulangerie, pâtisserie, boucherie, traiteur mais ils sont tout de même similaires : on y retrouve une salle principale avec une table, un espace lavage de mains, des échelles de stockage et les machines spécifiques au domaine afin de réaliser les produits qui seront vendus, des chambres froides pour stocker ses produits et des chambres de stockage à température ambiante pour ranger le matériel ou certains ingrédients.



Figure 8 : Salle principale d'un laboratoire



Figure 9 : Echelle et zone de lavage de mains d'un laboratoire



Figure 10 : Zone de stockage d'un laboratoire

Le laboratoire et le matériel nécessaire au scénario ayant déjà été modélisés sur Unity, je les ai principalement améliorés en modifiant certains objets ainsi que leurs animations tel que l'ouverture des portes des chambres froides ou des fours afin de les rendre plus réalistes.



Figure 11 : Salle principale d'un laboratoire sous Unity



Figure 12 : Zone de stockage d'un laboratoire sous Unity

## Le scénario

La procédure que nous allons suivre pour cette formation a été décrite dans un cahier des charges donné en annexes par un membre de l'équipe métier d'e-ManRisk. C'est une procédure de cuisson, refroidissement rapide et stockage d'un aliment qui est réalisée dans les laboratoires des magasins.

Il fallait donc réaliser à l'aide de ce cahier des charges des diagrammes UML que le système Mascaret allait pouvoir utiliser et analyser pour produire le scénario sous Unity. Mascaret parcourant les diagrammes, il faut être rigoureux lors de la réalisation de ceux-ci. Je vais principalement expliquer deux types de diagrammes que j'ai réalisés : le diagramme de classe et le diagramme d'activité.

Le premier diagramme réalisé est un diagramme de classe dans lequel nous allons modéliser tous les objets qui nous seront nécessaires pour la suite avec

- leurs attributs, ce sont des informations qui caractérisent un objet et qui pourront être utilisées comme conditions de passage à une prochaine étape, on doit également préciser le type de donnée que ce sera
- leurs méthodes, ce sont des opérations que l'objet peut réaliser
- le lien entre les différents objets

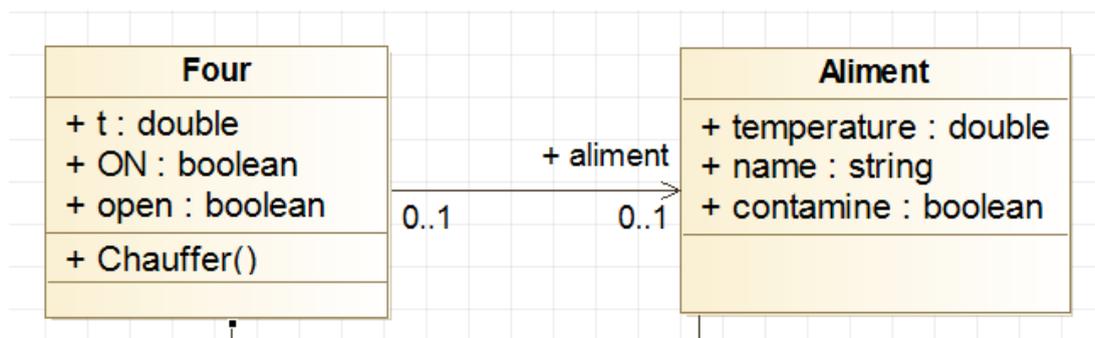


Figure 13 : Exemple d'un diagramme de classe

Dans cet exemple, le four possède 3 attributs : t, sa température, qui sera un nombre qui peut être à virgule et ON, s'il est allumé ou non, et open, s'il est ouvert ou non, qui seront à l'état vrai ou faux ainsi qu'une méthode chauffer(). Le four a également un lien avec un aliment car il peut ou non en contenir un.

Cette étape est très importante car c'est grâce à ce diagramme que nos objets sont caractérisés et c'est ce qui nous permettra d'avancer dans le scénario en posant des conditions sur les valeurs des attributs par exemple.

Le deuxième type de diagramme est un diagramme d'activité dans lequel sont modélisé :

- les étapes, qui sont soit des opérations qui seront codées par la suite, soit des explications faites généralement par le formateur
- les acteurs, qui la réalise modélisé par des conteneurs dans lesquels les étapes seront placées, ici le formateur ou l'apprenant
- leur ordre
- les objets nécessaires à leur réalisation (issue du diagramme de classe)
- les conditions, s'il y a plusieurs étapes après une et qu'il faut définir laquelle est la suivante

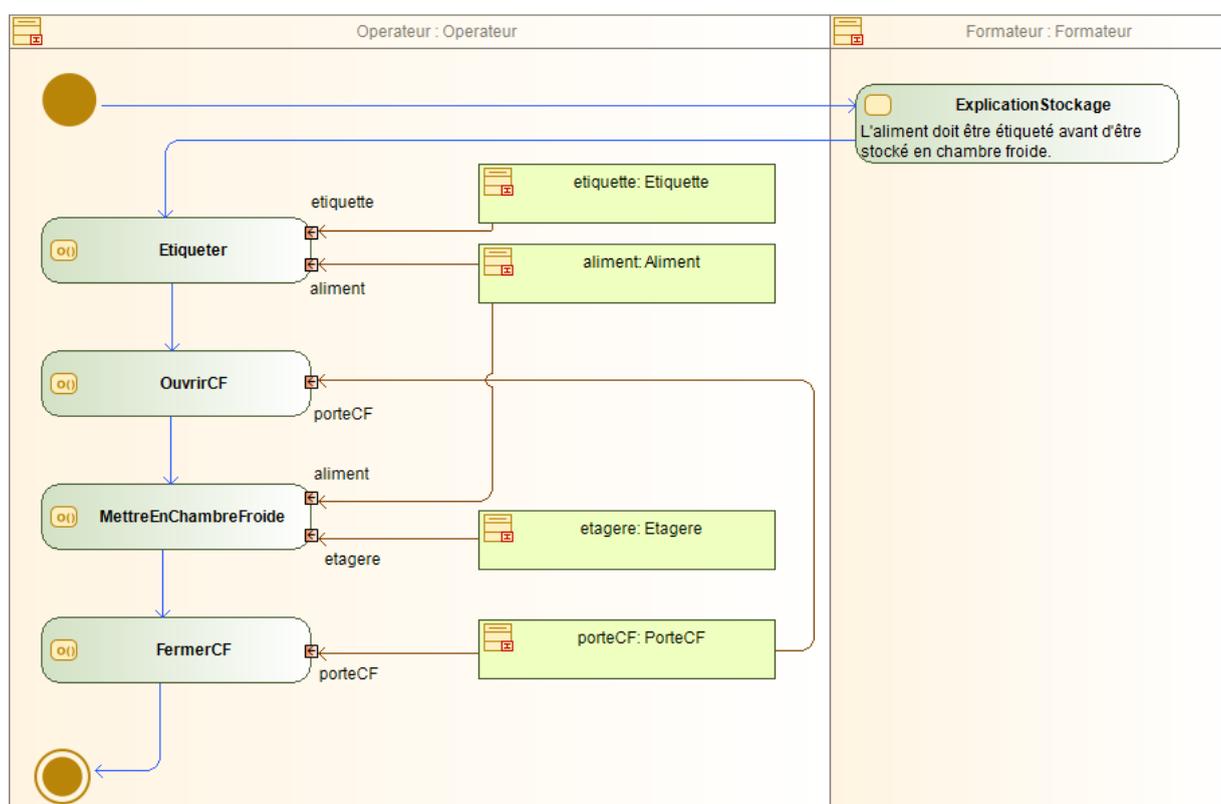


Figure 14 : Exemple d'un diagramme d'activité : stockage en chambre froide

Ce diagramme représente une procédure de stockage : on retrouve deux conteneurs représentant nos acteurs (un pour l'opérateur et un pour le formateur) avec une suite d'étapes. Nous n'avons pas de conditions ici, les étapes vont s'enchaîner dans l'ordre dans lesquelles elles sont reliées: le formateur va tout d'abord expliquer la procédure de stockage puis l'opérateur, qui est la personne suivant la formation, va devoir étiqueter l'aliment avec une étiquette, puis il va ouvrir la chambre froide, mettre l'aliment sur une étagère dans celle ci et pour finir la refermer.

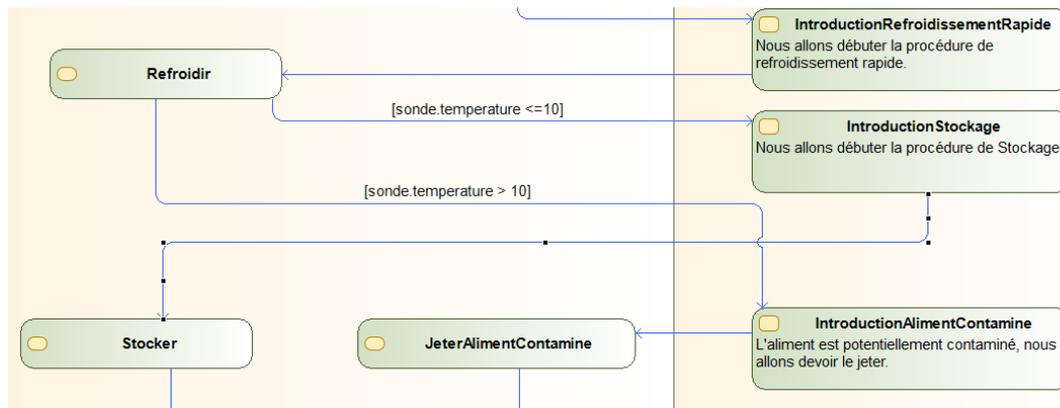


Figure 15 : Modélisation d'un choix dans un diagramme d'activité

Cet exemple montre la modélisation d'un choix dans un diagramme d'activité. Une fois que l'opération "refroidir" est finie, deux étapes suivent : l'introduction au stockage ou l'introduction au jet d'un aliment contaminé. Le choix de la prochaine action dépendra de la température de l'objet sonde: si cette température est supérieure à 10 degrés l'aliment doit être jeté alors que si la température est inférieure ou égale à 10 degrés l'aliment doit être stocké.

## Le lien entre l'environnement 3D et le scénario

Une fois la modélisation de mon scénario réalisée sous *Modelio*, il doit être exporté dans un fichier *XMI* qui sera parsé par Mascaret pour être joué sur Unity, pour que cela soit possible il faut créer un lien entre l'environnement 3D et le fichier *XMI* qui représente notre scénario. Ce lien va être réalisé par un fichier *XML* que l'on va créer.

D'une part, ce fichier va mettre en lien les objets créés dans le diagramme de classe avec ceux présents dans l'environnement virtuel en créant une entité dédiée.

```

<Entity name="Stove" class="Four">
  <Shape url="Stove"/>
  <Attribute name="t" value="20"/>
  <Attribute name="ON" value="false"/>
  <Attribute name="open" value="false"/>
</Entity>
  
```

Figure 16 : Création d'entité

On crée ici une entité nommée "Stove" qu'on associe à la classe four de notre diagramme, on indique ensuite qu'elle correspond à un objet dans le monde 3D qu'on a appelé "Stove" en utilisant la balise "Shape". On retrouve également les attributs qui étaient dans le diagramme de classe et on peut leur donner une valeur initiale, dans cet exemple, on fixe la température t de notre four à 20 degrés, le four est éteint et fermé donc les attributs ON et open ont comme valeur "false".

Il faut maintenant affecter les entités aux objets qui ont été utilisés dans les diagrammes d'activité.

```
<Affect type = "Ressource">
  <Ressource>four</Ressource>
  <Entity>Stove</Entity>
</Affect>
```

Figure 17 : Affectation d'entité

Pour ce faire, on utilise des balises "Affect" où l'on indique d'une part la ressource qui est l'objet que l'on a utilisé dans le diagramme d'activité puis l'entité qui correspond. Ceci nous permettra d'avoir accès à nos objets et à leurs attributs lorsque l'on va coder les étapes que l'on a défini dans les diagrammes.

## L'implémentation des étapes

L'implémentation des étapes se fait en C# car c'est le langage utilisé par le moteur Unity et Mascaret. Lorsque l'on arrive à une étape, la fonction "execute" de celle-ci est appelée.

```
36  override public double execute(double dt)
37  {
38      if (start == 0)
39      {
40          Debug.Log("##### Tu es dans l'action Operateur_OuvrirFour ");
41
42          GameObject mas = GameObject.Find("MascaretApplication");
43          uma = mas.GetComponent<UnityMascaretApplication>();
44          uma.help.text = "---Execute OP Operateur_OuvrirFour";
45          uma.text.text = "Il faut ouvrir le four";
46
47          uma.GetComponent<putInRed>().putInRedOnlyOne(GameObject.Find("Stove/Door"));
48
49          start++;
50      }
51      if (four.getProperty("open").getValue().getBoolFromValue()==true)
52      {
53          Debug.Log("----Condition passé");
54
55          uma.GetComponent<putInRed>().endPutInRed();
56          uma.text.text = "";
57
58          start--;
59
60          return 0.0f;
61      }
62      return 0.2f;
63  }
```

Figure 18 : Fonction execute() de l'opération "OuvrirFour"

Cette fonction va être appelée tant qu'elle ne retourne pas 0.0f. On va donc l'utiliser d'une part pour communiquer avec l'apprenant et pour définir la condition de passage à l'étape suivante.

D'une part, la première boucle "if" (ligne 38) va nous permettre d'indiquer dans l'environnement virtuel ce qu'il faut faire à cette étape :

- ligne 45 : on indique à l'apprenant par une phrase simple l'action à réaliser, ici "Il faut ouvrir le four"
- ligne 47 : on met en évidence la porte du four pour aider l'apprenant



Figure 19 : Indication pour l'opération "OuvrirFour"

La valeur de start est ensuite incrémentée à la ligne 49 pour que l'on ne repasse pas dans cette boucle.

La deuxième boucle "if" (ligne 51), nous sert à attendre la validation de l'étape, le code qui la suit n'est exécuté que si la valeur de l'attribut open du four est égale à "true" c'est-à-dire si le four est ouvert ce qui est bien le but de notre étape. Dans ce cas :

- ligne 55 : on arrête de mettre la porte du four en rouge
- ligne 56 : on enlève l'indication écrite
- ligne 60 : on retourne 0.0f, Mascaret va donc passer à l'étape suivante indiquée par le diagramme d'activité et appeler sa fonction "execute"

## 5.4. La gestion du temps

La procédure que l'apprenant est amené à réaliser au cours de cette formation peut être assez longue, il y a le temps de la cuisson mais aussi le temps du refroidissement qui peut durer jusqu'à deux heures. Dans le cadre réel, un opérateur serait amené à réaliser d'autres procédures ou à la réaliser avec d'autres aliments pendant que l'aliment cuit ou refroidit, dans le cas d'une formation cela ne sert à rien de surcharger l'apprenant avec d'autres aliments ou plusieurs procédures, il fallait donc trouver un moyen de faire s'écouler le temps au sein de l'environnement virtuel sans que la cuisson ou le refroidissement ne paraissent immédiat. Deux solutions pour spatialiser l'environnement dans le temps ont donc été mises en place : l'utilisation d'horloge et l'utilisation d'écran noir.

### Les horloges

Une horloge a été mise dans l'environnement afin d'indiquer l'heure à l'apprenant. Tout au long de la formation, le temps passe à un rythme normal et est affiché sur celle-ci. Une deuxième horloge est placée au-dessus du lave-main pour indiquer le temps que l'apprenant prend à se laver les mains et faire respecter le fait que cette étape doit durer plus de 30 secondes.

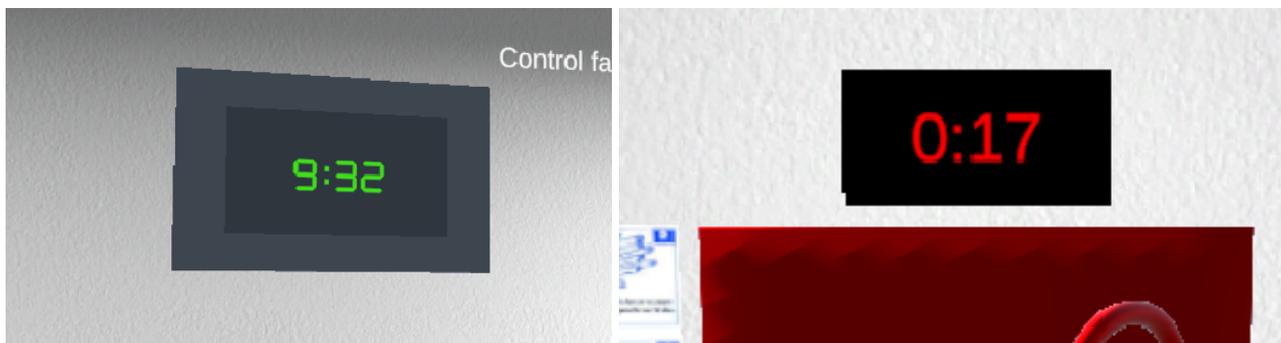


Figure 20 : Les horloges dans l'environnement virtuel

### Les écrans noirs

L'utilisation d'écran noir pour faire passer le temps est un procédé très utilisé que ce soit dans les jeux ou les dessins animés. J'ai décidé d'en utiliser lorsque le temps doit s'écouler d'un coup, par exemple lorsque l'apprenant met le poulet dans le four et que celui doit cuire, un écran noir avec le temps qui passe s'affiche pendant quelques secondes. Ce temps passe aussi sur l'horloge principale pour que cela reste cohérent.

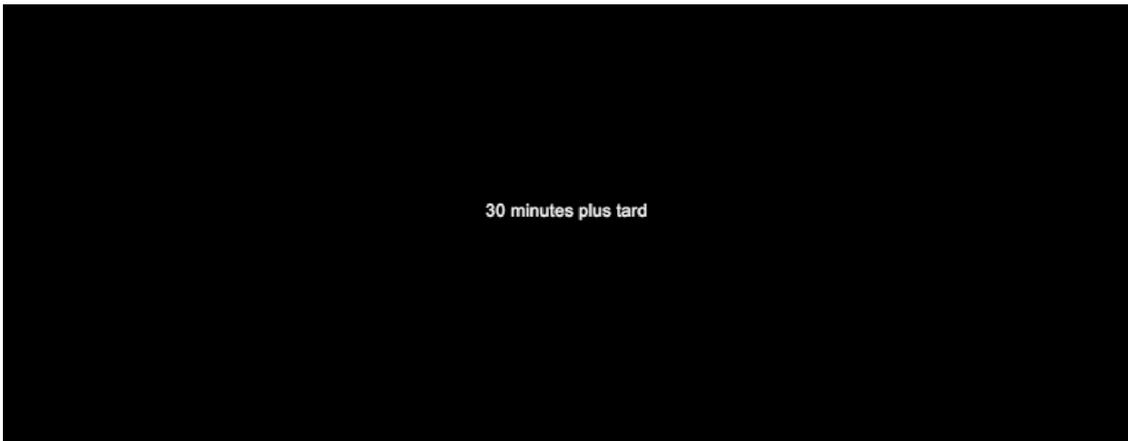


Figure 21 : Ecran noir pour passer le temps

## 5.5. La gestion de la contamination alimentaire

Cette formation s'intéressant principalement à l'hygiène alimentaire et cette procédure ayant pour but de savoir si un aliment a été potentiellement contaminé par des bactéries, il est important que la contamination alimentaire soit au cœur de notre réalité virtuelle, on va donc la rendre visible pour que l'apprenant la perçoive mieux et le confronter au cas où l'aliment est contaminé ou non.

### Visualisation des bactéries

Afin que l'apprenant comprenne mieux les dangers des bactéries, on va appliquer un aspect différent à l'aliment quand il est contaminé ainsi qu'aux mains lorsqu'elles sont sales. Cette visualisation permet de rendre plus explicite la contamination alimentaire et est plus ludique et marquante que de simplement dire que les mains sont sales ou que l'aliment est contaminé.

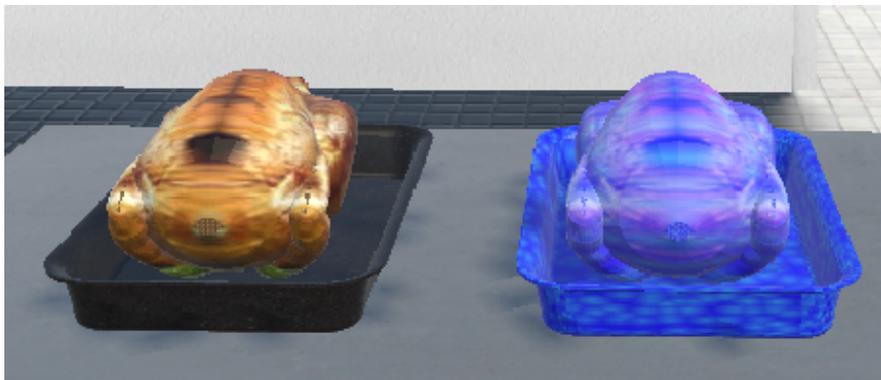


Figure 22 : Aliment non contaminé/contaminé

## L'impact des bactéries dans la formation

En réalité, les opérateurs ne savent pas si un aliment est contaminé ou non avant d'arriver à la fin de la procédure de refroidissement rapide, c'est pourquoi le choix a été fait de faire pareil dans cette formation : une valeur aléatoire est attribuée à l'aliment au lancement de la simulation afin de fixer s'il est contaminé ou non, et l'apprenant ne le saura que lorsque la procédure de refroidissement sera finie et qu'il devra stocké ou non l'aliment.

Mais la contamination alimentaire passe aussi par un mauvais respect des procédures, l'apprenant commence sa formation avec les mains sales comme cela serait le cas dans la vraie vie, ainsi s'il touche un aliment avant de s'être correctement lavé les mains ou si un aliment touche le sol celui ci va devenir contaminé, quelques secondes plus tard un écran disant au participant que l'aliment est contaminé et qu'il doit recommencer sera affiché.

### 5.6. Mise en place du logiciel d'e-ManRisk dans l'environnement virtuel

Le logiciel développé par e-ManRisk permet d'enregistrer le résultat des contrôles de qualité réalisés dans les laboratoires. Le logiciel est utilisé sur des tablettes car c'est ce qui est le plus pratique dans ce cadre. Un élément important de ce stage était donc de pouvoir faire de même dans la réalité virtuelle : avoir une tablette qui permettrait de prendre en main le logiciel et d'enregistrer le contrôle réalisé durant la formation.

Unity ne propose pas de navigateur web intégrable au sein de l'environnement 3D et il n'y a pas d'outil gratuit permettant ceci et créer ce type d'outil aurait été un travail trop volumineux, il fallait trouver une autre solution pour avoir accès à l'application d'e-ManRisk. Cette solution a été trouvée lors du précédent stage : il s'agit d'utiliser la librairie Puppeteer.

La tablette ayant déjà été modélisée et la librairie Puppeteer n'ayant pas encore totalement été mise en place, mon travail sera donc d'intégrer cette librairie à l'environnement virtuel.

### La librairie Puppeteer

La librairie Puppeteer permet d'interagir avec un navigateur web Chrome, via un serveur Node, comme on le ferait sur un ordinateur de manière classique mais de manière automatisée. On peut ainsi simuler des clics de la souris ou des appuis sur les touches du clavier mais également extraire des pdf ou faire des captures d'écran de page.

C'est ainsi que l'on naviguera sur l'application web d'e-ManRisk : une capture d'écran de la page actuelle sera prise et affichée sur la table dans l'environnement virtuel, l'utilisateur devra être en mesure de cliquer sur le site et la page affichée devrait changer ou des éléments devraient apparaître. On souhaite également que les champs textuels puissent être remplis, il faudra donc pouvoir modéliser un clavier pour la tablette.

La première étape a été de créer une architecture serveur-client qui utilisera la librairie Puppeteer.

## L'architecture serveur-client

L'architecture serveur-client est un mode de communication en réseaux entre différents acteurs qui sont de types serveur ou client, et qui utilise un protocole de communication défini et commun à tous les acteurs.

- Le client envoie des requêtes à un serveur, c'est lui qui engage la communication, et attend la réponse du serveur.
- Le serveur lui attend qu'un client lui envoie une requête, il est passif, une fois une requête reçue, il la traite et envoie une réponse.

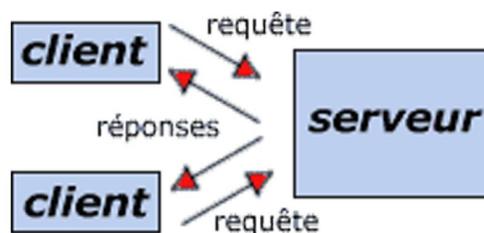


Figure 23 : Architecture serveur-client

Nous utiliserons le protocole *HTTP* pour notre architecture.

Nos communications suivront donc le schéma suivant :

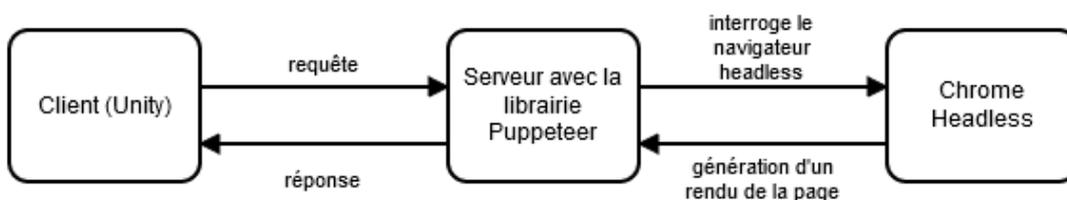


Figure 24 : Architecture serveur-client de notre situation

## Le protocole HTTP

Le protocole HTTP est un des protocoles de communication réseaux les plus utilisés.

Les requête et les réponses sont constitué de plusieurs lignes permettant de transférer les informations qui les concernent :

- une ligne de requête ou de statut (dans le cadre d'une réponse)
- une en-tête
- des données

Toutes ces informations sont formalisées, il est donc important de faire attention lors de la rédaction des requêtes car c'est une source fréquente d'échec de communications entre serveur et client.

La ligne qui nous intéresse le plus pour créer notre serveur est la première : la ligne de requête. Elle est composée de deux champs :

- la méthode est un verbe qui qualifie le type de requête
- une URL

Il existe plusieurs méthodes mais les plus utilisées sont GET et POST. La méthode GET sert à obtenir une information sur le serveur alors que la méthode POST sert à ajouter une information au serveur.

L'URL contient également plusieurs données :

- l'adresse du serveur
- la route du serveur que l'on souhaite utiliser
- optionnellement si la route en a besoin des paramètres ainsi que leurs valeurs

La route du serveur correspond à une fonction qu'il va exécuter pour pouvoir répondre, si cette fonction nécessite des paramètres supplémentaires il faut les indiquer.

```
VERB /adresse/route?paramètre1=valeur1&paramètre2=valeur2
```

Figure 25 : Format d'une requête HTTP

Il faut maintenant que nous définissions quelles routes le serveur aura besoin pour fonctionner.

## Le serveur

Le serveur que nous allons coder à pour but de simuler le navigateur web au sein de l'environnement 3D. Les requêtes qui lui seront adressées seront donc des actions que l'opérateur fera et pour lesquelles une action suivra avant que le serveur ne renvoie une capture d'écran du navigateur web.

Ainsi, le serveur comprendra 6 routes, 4 qui seront de méthodes GET et 2 de méthodes POST. Je vais détailler le but de chaque route ainsi que les paramètres dont elles auront besoin.

```
GET /adresse/init
GET /adresse/click?x=valueX&y=valueY
GET /adresse/up
GET /adresse/down

POST /adresse/fill?text=valueText
POST /adresse/select?optionNumber=n&x=valueX&y=valueY
```

Figure 26 : Route du serveur

La première route init sera une route d'initialisation qui permet de s'identifier à l'application d'e-ManRisk et qui renvoie une capture d'écran de la page d'accueil du site qui permet de choisir une activité.

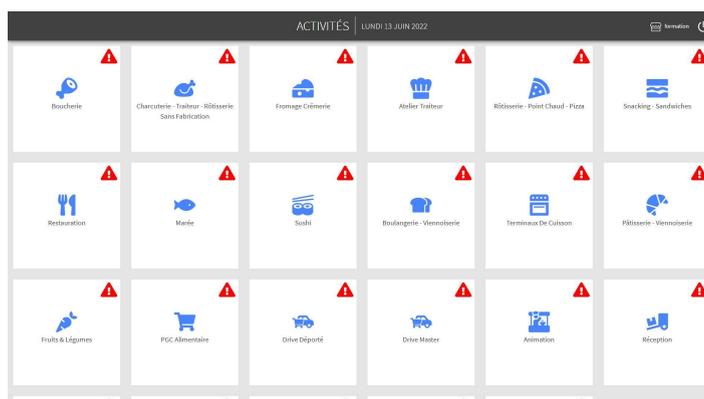


Figure 27 : Ecran d'accueil de l'application d'e-ManRisk

La route click va simuler un clic de souris aux coordonnées x et y spécifier de l'écran et va renvoyer une capture d'écran une fois que la page suivante aura été chargée ou que les éléments modifiés par le clic l'aient été. Elle renvoie également le type d'élément HTML qui a été cliqué, nous reviendrons sur cet élément lors de l'analyse de la réponse par le client.

Les routes up et down sont nécessaires car certaines pages de l'application sont plus grandes que le format de l'écran de la tablette, il faut donc que l'utilisateur puisse descendre ou monter dans la page. Une capture d'écran est ensuite prise et envoyée en réponse.

La route fill va permettre de simuler l'appuie sur des touches du clavier pour remplir des champs textuels sur la page et répondre avec la capture d'écran une fois les champs remplis.

La route select va permettre de sélectionner l'option numéro "n" d'une liste déroulante qui sera située aux coordonnées x et y d'une page et également envoyée une capture d'écran.

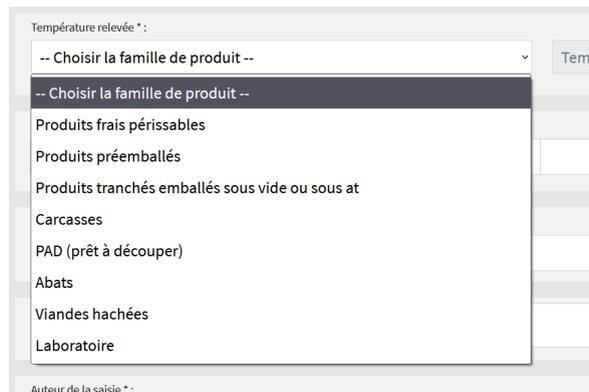


Figure 28 : Exemple de liste déroulante du site

Je vais décomposer le code de la route fill comme exemple :

```
170 Route.post('/fill', async (ctx) => {
171   console.log("\n---FILL---")
172
173   const queryElements = ctx.request.all();
174
175   await page.evaluate( () => {
176     document.activeElement.value = "";
177   },
178   { }
179 );
180
181 if (queryElements.text) {
182   let text = queryElements.text;
183
184   console.log("Je remplis avec: ", text);
185
186   await page.keyboard.type(text);
187
188   //Screenshot
189   const filename = Math.floor(Date.now() / 1000);
190   await delay(300);
191   await page.screenshot({ path: `${filename}.jpeg` });
192   console.log("screenshot");
193   const image = fs.readFileSync(`${filename}.jpeg`);
194   fs.unlinkSync(`${filename}.jpeg`);
195
196   return {
197     image: 'image/jpeg;base64,' + Buffer.from(image).toString('base64'),
198   };
199 }
200 }
201 });
```

Figure 29 : Exemple de code d'une route

Pour que cette requête fonctionne comme attendue, il faut qu'un champ de texte soit activé (c'est-à-dire que l'utilisateur ait cliqué dessus).

La requête attendue pour cette route doit bien être de méthode POST comme l'indique la ligne 170.

A la ligne 177, on efface ce qui été auparavant écrit dans ce champ. Puis on récupère le paramètre text de l'URL de notre requête grâce à la ligne 183 afin de pouvoir simuler le tapage de ce texte au clavier à la ligne 187.

Une capture d'écran de la page est ensuite faite à l'aide de la ligne 192, elle sera ensuite envoyée en réponse par les lignes 197/198.

## Le client

Le client de notre serveur sera donc notre environnement de réalité virtuelle, le moteur de jeux Unity possède un module nommé UnityWebRequest qui nous permettra d'envoyer les requêtes au serveur.

Voici le code permettant d'envoyer la requête fill à notre serveur :

```
514 public IEnumerator Fill(string t)
515 {
516     Debug.Log("---Fill---");
517     Debug.Log("txt: " + t);
518
519     WWWForm form = new WWWForm();
520
521     string pos = "http://127.0.0.1:3333/fill?text="+t;
522
523     UnityWebRequest www = UnityWebRequest.Post(pos,form); //méthode Get avec l'url voulue
524
525     yield return www.SendWebRequest(); //envoi de la requête
526
527     if (!string.IsNullOrEmpty(www.error)) // on regarde s'il y a une erreur
528     {
529         Debug.Log(" ----> ERREUR !");
530         Debug.Log(www.error); // on affiche l'erreur
531         request_decrypt = true;
532     }
533     else
534     {
535         Debug.Log(" ----> SUCCES !");
536         Debug.Log(www.downloadHandler.text); // on affiche la réponse
537         string response = System.Text.Encoding.UTF8.GetString(www.downloadHandler.data);
538         AnalyseHTTP screen = new AnalyseHTTP();
539         screen = JsonConvert.DeserializeObject<AnalyseHTTP>(response);
540
541         char[] separator = { ',', ' ' };
542         String[] strlist = screen.image.Split(separator, StringSplitOptions.RemoveEmptyEntries);
```

Figure 30 : Exemple d'envoi d'une requête avec Unity

Une action de l'opérateur appellera la fonction fill qui enverra à l'aide du module UnityWebRequest une requête de méthode POST à la ligne 523. L'URL de cette requête est défini ligne 521 et on retrouve bien *adresse/fill?text=t*.

La suite de la fonction permet de savoir s'il y a eu une erreur ou non lors de la communication et d'agir en conséquence. S'il n'y a pas d'erreur, la réponse est analysée afin que l'on puisse l'utiliser dans l'environnement virtuel. Je détaillerai l'analyse et l'utilisation de ces réponses dans une prochaine partie.

### La tablette dans l'environnement 3D

Dans un souci de précision pour interagir avec la tablette, il y aura en fait deux tablettes dans le monde virtuel : une première sera disposée dans le laboratoire à taille réelle que l'utilisateur devra saisir pour être transféré devant une plus grande tablette avec laquelle il pourra interagir par un pointeur.

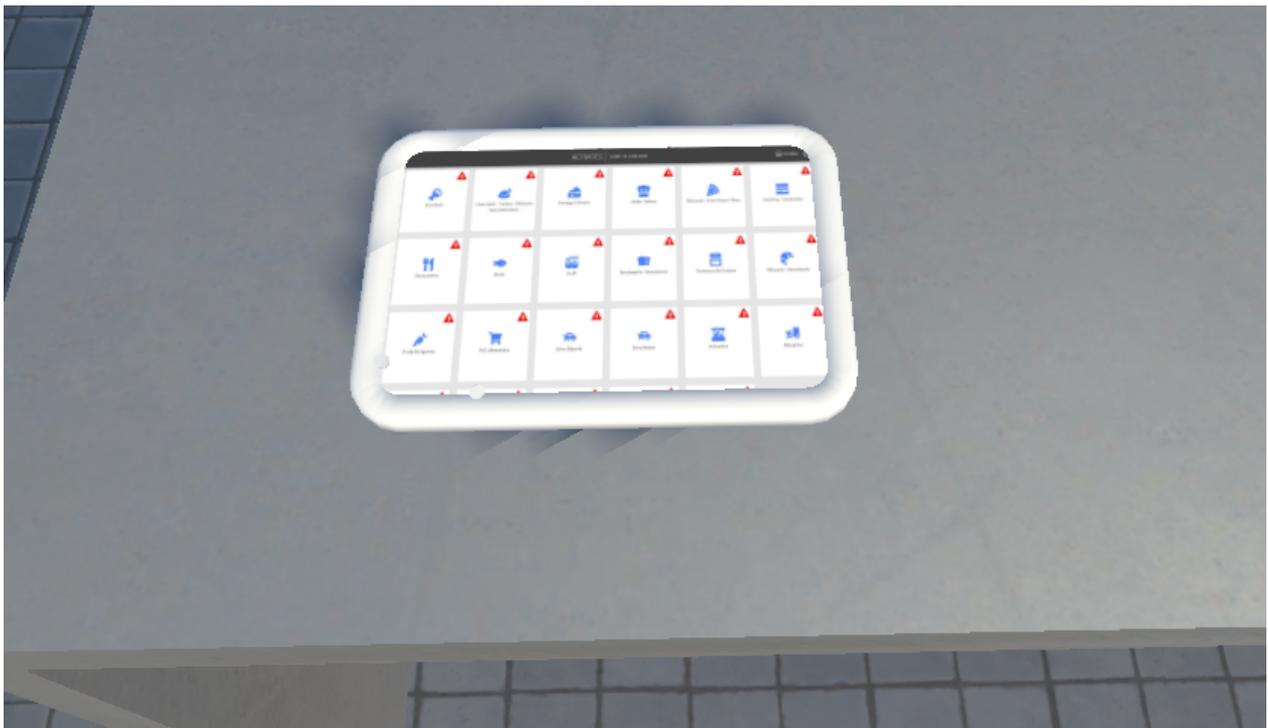


Figure 31 : Tablette dans le laboratoire

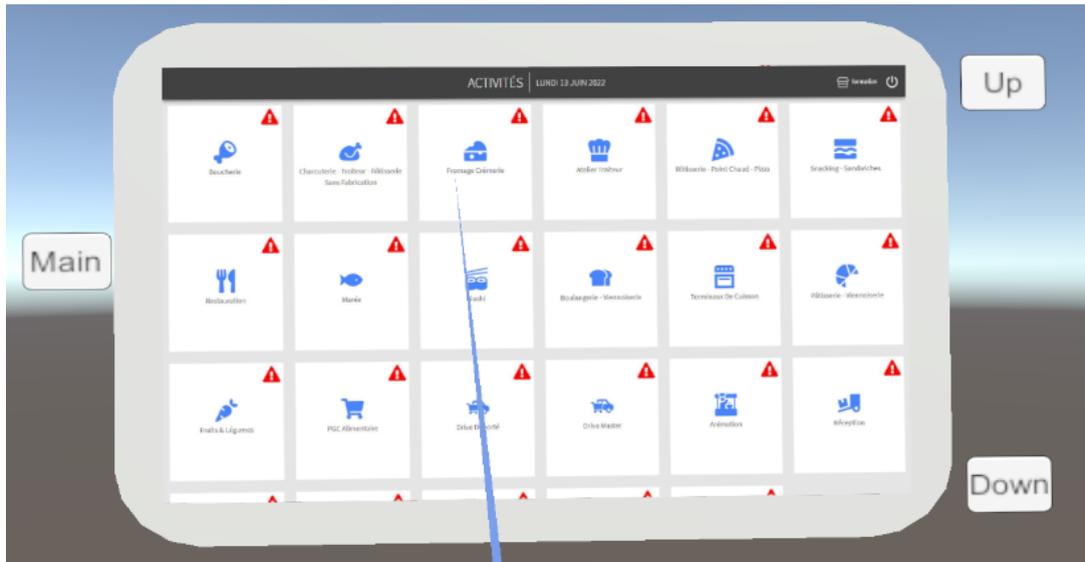


Figure 32 : Tablette interactive avec un pointeur

L'image qui est envoyée par le serveur en réponse des requêtes doit donc être affichée sur ces deux tablettes, le code permettant d'inclure la capture d'écran est similaire quelque soit la route empruntée.

```

418     Debug.Log(" ----> SUCCES !");
419     Debug.Log(www.downloadHandler.text); // on affiche la réponse
420     string response = System.Text.Encoding.UTF8.GetString(www.downloadHandler.data);
421
422     Debug.Log("rep: "+response);
423     //On récupère le screenshot
424     AnalyseHTTP rep = new AnalyseHTTP();
425     rep = JsonConvert.DeserializeObject<AnalyseHTTP>(response);
426
427     char[] separator = { ',', ' ' };
428     String[] strlist = rep.image.Split(separator, StringSplitOptions.RemoveEmptyEntries);
429     //Debug.Log("strlist: " + strlist[0]);
430
431     byte[] imageBytes = Convert.FromBase64String(strlist[1]); // on le converti en bytes
432     Texture2D tex = new Texture2D(2, 2);
433     tex.LoadImage(imageBytes);
434
435     //petite
436     GameObject imageScreen = GameObject.Find("Tablet/Canvas/tabletteScreen");
437     Debug.Log(imageScreen);
438     imageScreen.GetComponent<RawImage>().texture = tex;
439     //grande
440     imageScreen = GameObject.Find("/TabletScene/Tablet/Canvas/tabletteScreen");
441     imageScreen.GetComponent<RawImage>().texture = tex;

```

Figure 33 : Récupération des captures d'écran

Lorsqu'il n'y a pas d'erreur de communication entre le serveur et le client comme indiqué ligne 418, le client (ici Unity) récupère la réponse avec la ligne 420 et en extrait l'image grâce à la ligne 431. Cette image sera ensuite placée sur les écrans des deux tablettes sous la forme d'une texture aux lignes 438 et 441.

## L'interaction avec la tablette dans l'environnement 3D

L'appuie sur la gâchette de la manette du casque de réalité virtuelle en pointant le bouton Up ou Down déclenche l'envoi respectivement de la requête up ou down. Au contraire, si l'utilisateur pointe sur l'écran de la tablette la requête click sera envoyée avec la position du pointeur. Comme indiqué précédemment, la requête click renvoie aussi le type d'élément HTML, cette information est nécessaire pour permettre l'affichage d'un clavier qui permettra de remplir les champs textuels de l'application.

```
443 //On cherche à savoir si il faut le clavier
444 char[] separatorClavier = { " ", "" };
445 String[] elementTagName = rep.info.elementTagName.Split(separatorClavier, StringSplitOptions.RemoveEmptyEntries);
446 Debug.Log("-->" + elementTagName[0] + "--");
447 if(elementTagName[0] == "INPUT")
448 {
449     GameObject.Find("Keyboard").GetComponent<Clavier>().ChangeState("Number", null , rep.info.actualText);
450 }
451 else if(elementTagName[0]=="TEXTAREA")
452 {
453     GameObject.Find("Keyboard").GetComponent<Clavier>().ChangeState("Clavier", null , rep.info.actualText);
454 }
455 else if(elementTagName[0]=="SELECT")
456 {
457     List<string> selectOptionsText = rep.info.selectOptionsText;
458     Debug.Log("---SELECT---");
459     Debug.Log(selectOptionsText[0]);
460     GameObject.Find("Keyboard").GetComponent<Clavier>().ChangeState("Options",selectOptionsText);
461 }
462 else
463 {
464     GameObject.Find("Keyboard").GetComponent<Clavier>().ChangeState("None");
465 }
466 request_decrypt = true;
467
```

Figure 34 : Récupération du choix du clavier

On constate que selon l'élément sur lequel on a cliqué l'effet ne sera pas le même :

- "INPUT" (ligne 447) nécessite d'afficher un clavier de nombre (ligne 449)
- "TEXTAREA" (ligne 451) nécessite d'afficher un clavier avec des lettres (ligne 453)
- "SELECT" (ligne 455) nécessite d'afficher différentes options (ligne 457)
- sinon (ligne 463) il n'y a pas besoin d'afficher de clavier (ligne 465)

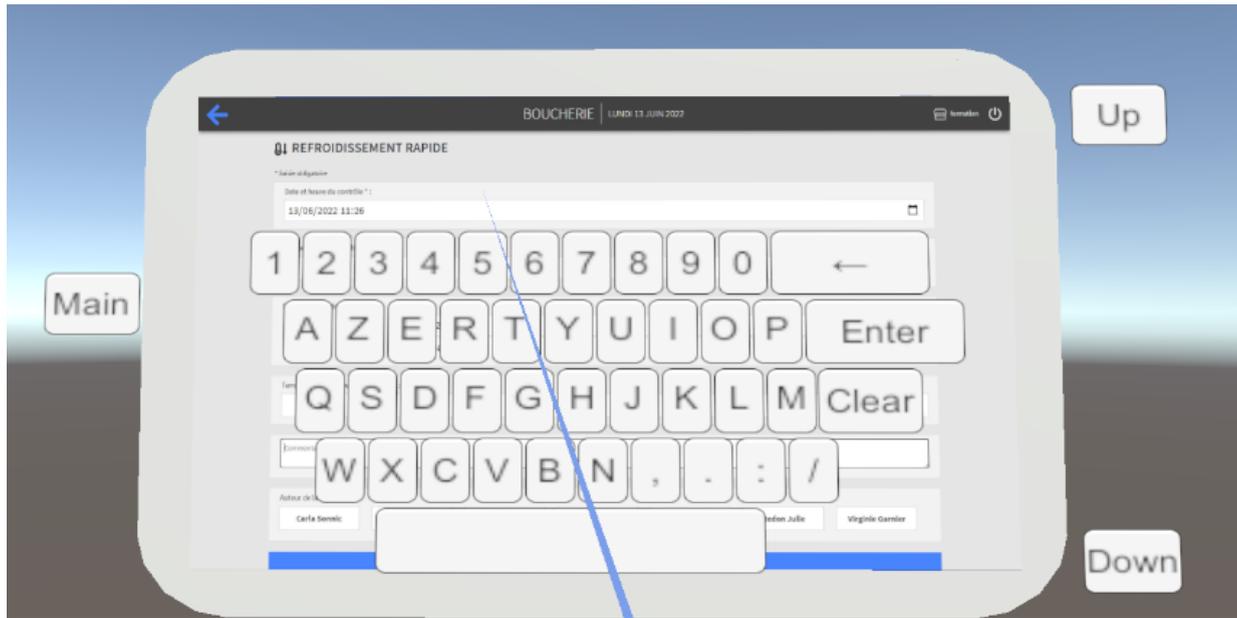


Figure 35 : Affichage du clavier

Ainsi, à l'aide du clavier l'utilisateur va pouvoir remplir le formulaire du scénario, une fois sa réponse inscrite l'appui sur "Enter" va envoyer la requête fill avec comme texte ce qui a été saisi.

Tous ces éléments permettent une utilisation complète de l'application développée par e-ManRisk au sein de l'environnement de réalité virtuelle. On peut donc compléter un formulaire, il sera ainsi envoyé sur le serveur et accessible de manière classique depuis l'application sur une vraie tablette ou un navigateur web.

## 6. Perspectives futures du projet

### 6.1. La formation en Réalité Virtuelle chez e-ManRisk

Ce projet, ayant pour but de permettre à l'entreprise e-ManRisk d'avoir un aperçu de ce que pourrait être la formation en réalité virtuelle, les a conquis. En effet, la réalisation de formation 3D vient s'ajouter à la liste des projets que l'entreprise souhaite mettre en place dans les prochaines années, ce serait une innovation au sein du mouvement E. Leclerc personne ne propose cela pour l'instant.

De plus, ce projet est un prototype, de nombreuses améliorations sont réalisables : modélisation d'un humain pour le formateur, gérer par de l'Intelligence artificielle ou par une vraie personne, utilisation de Text-To-Speech. Le type de formation peut également être modifié et on peut ajouter d'autres scénarios afin de les industrialiser.

### 6.2. Les tests fonctionnels front end

Mon travail concernant la librairie Puppeteer dans le cadre ce projet, pourra par la suite être utilisé par l'entreprise dans un tout autre contexte : la mise en place de tests fonctionnels front-end pour ses applications web.

## 7. Bilan personnel

Je n'avais pour le moment pas encore eu d'expériences professionnelles dans le domaine de l'informatique qui est celui vers lequel je souhaite m'orienter dans le futur.

### 7.1. Bilan des compétences

Cette expérience m'a apporté beaucoup de compétences utiles pour mon futur.

Tout d'abord la modélisation du scénario sous la forme de diagramme d'activité et l'utilisation de diagramme de classe pour tous les éléments nécessaires ont entraînés ma rigueur car il ne fallait pas faire d'erreur sinon Mascaret ne pouvait fonctionner par la suite.

De plus, la modélisation UML étant un outil important dans le domaine de l'informatique et l'ayant seulement jusqu'à présent utilisé dans le domaine scolaire, la mise en place de celle-ci au sein de mon stage m'a permis de mieux l'assimiler.

Il en est de même pour la programmation orientée objet que je n'avais utilisée que dans le cadre des cours.

De plus, ce stage m'a permis de prendre en main le moteur de jeux Unity et le langage C#, qui me seront utiles dans le futur.

### 7.2. Orientation futur

Réalisant à partir de la rentrée d'août 2022, un double diplôme : "Maîtrise en informatique option Jeux Vidéo" à l'UQAC, ce stage m'a permis de confirmer mon choix d'orientation futur et m'a éclairée dans le choix des modules que je vais prendre pour ma maîtrise qui s'articulent autour du jeu vidéo, du développement 3D et de la VR mais aussi des jeux sérieux.

## 8. Conclusion

Pour conclure, ce stage m'a apporté beaucoup, que ce soit sur le plan social ou sur le plan technique, comme détaillé plus tôt, de plus, il a totalement répondu à mes attentes.

Il m'a également permis de découvrir le monde du travail dans le domaine de l'informatique et de me conforter dans mes choix de modules pour ma dernière année d'étude que je réaliserai en double diplôme : "Maîtrise en informatique option Jeux Vidéo" à l'UQAC et pour mes choix de carrières futures.

Je me suis réellement épanouie au sein des équipes d'e-ManRisk, cette expérience au sein d'une filiale à échelle humaine d'un grand groupe m'a permis d'en apprendre plus sur moi-même et mes envies futures en termes de tailles des équipes avec lesquelles je souhaite travailler et de type d'entreprise que je souhaite intégrer.

## 9. Bibliographie

**Documentation Unity :** <https://docs.unity3d.com/ScriptReference/index.html>

**Documentation SteamVR :**

[https://valvesoftware.github.io/steamvr\\_unity\\_plugin/api/index.html](https://valvesoftware.github.io/steamvr_unity_plugin/api/index.html)

**Documentation Puppeteer :** <https://pptr.dev/>

**Documentation Web API :** <https://developer.mozilla.org/en-US/docs/Web/API>

**Documentation Camunda BPMN :** <https://camunda.com/bpmn/reference/>

## 10. Annexes

### 10.1. Scénario du projet

Objet	Description d'un scénario dans le cadre du projet RV
Date	Le 11 mars 2021
Participants	S/O

Version	Auteur	Date	Description
1.0	Nicolas SIMON	11/03/2021	Création initiale

#### Résumé

Le projet consiste à construire un environnement virtuel qui vise à proposer à nos clients un support original de formation à l'hygiène alimentaire en mode Réalité virtuelle.

Le but de ce document est de rédiger un premier scénario en décrivant l'environnement ainsi que son déroulement qui sera proposé aux personnes formées.

Il s'agit d'intégrer le salarié dans un environnement de travail similaire au sien en lui proposant une série d'actions et leurs conséquences (conformes ou non-conformes) dans le cadre de la maîtrise de la sécurité alimentaire.

Pour ce premier scénario, je propose d'amener l'opérateur dans une opération de cuisson, de refroidissement rapide et de stockage.

#### Détail du scénario

Tâche : l'opérateur se verra confier quatre missions principales :

- Le lavage des mains
- La cuisson
- Le refroidissement d'un plat
- Le stockage de ce plat dans une chambre froide

Locaux visités :

- Laboratoire de fabrication
- Couloir
- Chambre de stockage

Matériels nécessaires :

- Lave main à commande non manuelle – distributeur à savon et à essuie main
- Boîte de gants
- Bac gastro et son couvercle
- Poubelle à pédale

- Thermomètre à sonde, bouton ON/OFF, afficheur
- Echelle de stockage
- Armoire de refroidissement
- Housse de protection
- Étiquette balance vierge
- Marqueur

### Etape 1 : le lavage des mains

Avant toute opération de fabrication, l'opérateur doit se laver les mains selon des règles très précises.

Matériel nécessaire :



- ✓ **Scénario idéal** : utilisation du lave-main à commande non manuelle (allumage de l'arrivée d'eau à l'aide du genou par pression sur plaque frontale), une pression sur le distributeur de savon pour une dose puis lavage des mains sur une durée minimale de 30 secondes en respectant le protocole ci-dessous :



Après le lavage, le rinçage, encore une fois à l'aide de la commande non manuelle.  
Enfin le séchage par **tapotage** d'un papier à usage unique sur toute la surface lavée.  
Le papier à usage unique est ensuite jeté à la poubelle qui sera ouverte avec le pied.



Tous les autres scénarios possibles seront non conformes avec des degrés de gravité variés selon l'erreur commise. Suggestion : signifier l'erreur ou le manquement par un visuel fort (croix rouge ?) et faire comprendre l'erreur par une vision virtuelle du danger.

Ex : si l'opérateur se saisit d'un torchon que le scénario prévoit de laisser à disposition pour s'essuyer les mains (piège), prévoir une visualisation virtuelle des colonies de bactéries présentes sur le support interdit.

- ✓ **Scénario 1** : manipuler la commande d'eau non manuelle à la main
- ✓ **Scénario 2** : manquer l'une ou l'autre (ou plusieurs) des étapes du protocole de lavage des mains
- ✓ **Scénario 3** : se laver les mains en moins de 30 secondes
- ✓ **Scénario 4** : s'essuyer les mains avec un torchon
- ✓ **Scénario 5** : s'essuyer les mains (au lieu du tapotage)
- ✓ **Scénario 6** : ouvrir la poubelle pour jeter l'essui main avec les mains (le couvercle) sans utiliser la pédale.

## Etape 2 : cuisson

La cuisson a pour importance microbiologique de détruire les bactéries pathogènes.

Parmi celles-ci, la salmonelle est la plus thermo résistante. Mais elle-même, à 63°C, est détruite.

L'objectif de cette première étape est donc de s'assurer que le thermomètre affiche au minimum la température de +63°C.

Le plat est mis en cuisson dans une gamelle sur un piano. Le thermomètre est disposé dans l'aliment pour vérifier sa température de cuisson.

- ✓ **Scénario idéal** : l'opérateur doit
  - Enfiler des gants
  - Allumer le thermomètre
  - Désinfecter la sonde du thermomètre (avec une lingette désinfectante)
  - Plonger la sonde à cœur du produit
  - A l'issue de la cuisson, l'opérateur doit penser à désinfecter la sonde du thermomètre.

Jeter la lingette à la poubelle en utilisant la pédale (sans toucher le couvercle).

Matériel nécessaire (de gauche à droite : gants, lingettes désinfectantes, thermomètre à sonde).



Option virtuelle : symboliser la montée en température avec un thermomètre graphique qui symbolise le passage en zone saine (au-dessus de +63°C).

Symboliser la présence de bactéries en grand nombre sur certaines surfaces : la sonde du thermomètre, la commande non manuelle du lave main, le couvercle de la poubelle (présence +++).

Les scénarii

- ✓ Scénario 1 : arrêt de la cuisson avant la température cible, représentation virtuelle de la présence microbienne.
- ✓ Scénario 2 : pas de désinfection de la sonde du thermomètre
- ✓ Scénario 3 : pas de port de gants.

### Etape 3 : le refroidissement

Théorie microbiologique : obligation réglementaire de passer d'une température supérieure à +63°C à une température inférieure à +10°C en moins de 2 heures.

La zone thermique +63 à +10°C est critique en microbiologie et doit être franchie le plus rapidement possible pour minimiser le risque de contamination et de multiplication microbienne.

Une représentation virtuelle d'un thermomètre et la descente du mercure est envisageable avec un changement de couleur selon la tranche de température : > à +63°C (vert), entre +10 et +63°C (rouge), inférieure à +10°C (vert).

- ✓ Scénario idéal : l'opérateur doit
  - Placer un bac gastro (avec l'aliment) sur une échelle
  - Placer l'échelle dans l'armoire de refroidissement
  - Fermer la porte de l'armoire de refroidissement
  - Lancer le programme

- Attendre maximum deux heures
- Désinfecter la sonde du thermomètre de la cellule de refroidissement
- Planter la sonde à cœur du produit
- Attendre que la température se stabilise sur l'afficheur.
- Si cette température est inférieure à +10°C, la noter (sur tablette ?)
- En cas de présentation de l'outil tablette, présenter un formulaire de refroidissement rapide.
- Si la température est supérieure à +10°C et que les 2 heures ne sont pas écoulées, poursuivre le refroidissement.
- Si la température est supérieure à +10°C et que les 2 heures sont écoulées, jeter le produit (potentiellement contaminé).
- Récupérer l'échelle, noter la température affichée sur le thermomètre
- L'étape suivante (stockage) ne peut être réalisée que si T°C inférieure à +10°C en moins de 2 Heures.

Matériel nécessaire : (de gauche à droite : échelle, bac gastro, cellule de refroidissement).



- ✓ Scénario 1 : pas de désinfection de la sonde du thermomètre
- ✓ Scénario 2 : durée supérieure à 2H
- ✓ Scénario 3 : température supérieure à +10°C et stockage de la marchandise
- ✓ Scénario 4 : température et durée non saisies sur support Tablette.

#### **Etape 4 : le stockage en chambre froide.**

Les choses importantes à savoir pour cette quatrième étape :

- Le froid conserve car limite le développement microbien sans le stopper.
- Comme il y a toujours multiplication bactérienne, la salubrité du produit a une limite de temps : c'est sa DLC à indiquer au marqueur sur une étiquette balance par exemple.

- Tout produit présent en chambre froide doit être identifié (quoi, fabriqué quand et DLC) et protégé (film, housse d'échelle).

Matériel nécessaire : (de gauche à droite : étagères de chambre froide, housse d'échelle, rouleau d'étiquettes vierges de balance).



QualityBooks

- ✓ **Scénario idéal** : l'opérateur doit
  - Protéger le bac ou l'échelle
  - Indiquer sur une étiquette nom, date de fabrication et/ou DLC
  - La stocker dans la chambre froide.
  
- ✓ Scénario 1 : stockage à nu
- ✓ Scénario 2 : une ou plusieurs indications manquantes sur le bac
- ✓ Scénario 3 : pas de stockage en CF.

## 10.2. Les outils utilisés

Je vais dans cette annexe présenter les outils que j'ai utilisés tout au long de mon stage.

### Camunda



Camunda est une plateforme open source d'automatisation des flux de travail et des décisions depuis 2013. Camunda gère notamment les diagrammes BPMN, qui sont un type de diagramme pour modéliser des processus métier que j'ai découvert et appris à utiliser au cours de mon stage.

### Gitlab



GitLab est une plateforme de développement collaborative open-source qui m'a permis de sauvegarder mon travail et de le versionner.

### Mascaret



Mascaret est un système développé par le CERV (Centre Européen de Réalité Virtuelle) pour la scénarisation et la gestion de formation en Réalité Virtuelle. C'est cet outil qui m'a permis d'intégrer le scénario de formation à la réalité virtuelle.

### Modelio



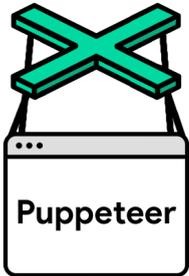
Modelio est un outil UML open-source développé par Modeliosoft depuis 1991. J'ai réalisé tous mes diagrammes UML à l'aide de ce logiciel.

### Postman



Postman est une application permettant de tester des APIs, créée en 2012. Cet outil m'a permis de tester mon serveur avant de l'intégrer à Unity.

## Puppeteer



Puppeteer est une librairie Node qui fournit une API pour contrôler une instance du navigateur Chrome Headless. Le langage Javascript est utilisé pour interagir avec le navigateur et permet de prendre des captures d'écran, de scraper le contenu de site web ou de faire des tests d'UI par exemple. Je l'ai utilisé pour avoir accès à l'application d'e-ManRisk.

## SteamVR



# Steam® VR

Steam VR est un outil développé par Valve Corporation qui permet d'avoir accès à de nombreux jeux en réalité virtuelle sur tous types de casque. Mais il propose aussi un plug-in et une API rendant l'intégration de la VR au sein du logiciel Unity plus simple.

## Unity



Unity est un moteur de jeux multi plateforme développé par Unity Technologies depuis 2005 et qui utilise principalement le langage C#. Il est l'un des plus répandus dans l'industrie du jeu vidéo. Il a été mon outil principal durant ce stage pour concevoir et améliorer l'environnement 3D.

## Visual Studio Code



# Visual Studio Code

Visual Studio Code est l'éditeur de code que j'ai utilisé tout le long de mon stage, que ce soit avec le langage C# pour Unity, avec XML pour l'intégration de mascaret ou en Node ou Javascript pour le serveur.