|  |  |
| --- | --- |
| **Événement** | Hackathon 2023 |
| **Lieu** | Université du Québec à Chicoutimi  555 Boulevard de l’Université  Chicoutimi (Québec) G7H 2B1 |
| **Thème** | Épreuve d’Intelligence artificielle : Explorations de solutions pour découvrir un chemin de cadenassage optimisé |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## Sujet de l’épreuve

CONFORMiT exerce dans le domaine de ce que l’on appelle la SSE « Santé, Sécurité, Environnement » (ou Environmental Health and Safety (EHS) en anglais). Ce domaine comporte de nombreux défis mais l’un des plus critiques est le **cadenassage** (lockout). Il s’agit, dans l’industrie Canadienne, de la procédure par laquelle on souhaite s’assurer qu’une machine industrielle a été correctement désactivée afin d’y effectuer une opération de maintenance ou d’entretien. Pour cela, on doit en général accéder à une série de dispositifs divers (valves, moteurs, convoyeurs, …) qui sont reliés à la machine principale, et y apposer des cadenas qui bloquent physiquement l’actionnement des dispositifs, et qui ne peuvent être enlevés que par la personne qui les as installés en premier lieu. De cette manière, on peut garantir qu’un dispositif reste verrouillé durant toute l’opération de maintenance, et qu’aucun collaborateur travaillant au sein de la machine principale ne risque de se trouver au mauvais endroit lors de la réactivation de la machine. Effectuer un bon cadenassage est donc **un enjeu crucial** des procédures industrielles.

La procédure de cadenassage peut très facilement augmenter en complexité et comporter un grand nombre d’étapes. Les **dispositifs connexes** qu’il est nécessaire de verrouiller pour accéder à une machine principale sont souvent nombreux et peuvent se trouver de part et d’autre d’une grande usine, il est donc crucial d’optimiser l’ordonnancement des étapes nécessaires afin de minimiser le temps passer pour sécuriser chaque machine.

Aujourd’hui, vous allez affronter une épreuve d’Intelligence Artificielle pour trouver un moyen d’optimiser le nombre d’étapes nécessaires pour **cadenasser complètement une machine**. Pour ce faire, nous allons vous fournir un environnement numérique représentant une version très simplifiée d’une usine, où se trouvent plusieurs **valves** ainsi qu’une **machine principale** (goal) à cadenasser. Votre mission va être de mettre au point, par la méthode de votre choix, un outil qui sera capable de trouver le (ou l’un des) **meilleur(s) chemin(s)** pour cadenasser toutes les valves **avant de cadenasser la machine principale.** Votre outil devra être performant indépendamment de la disposition et la taille de l’usine.

L’environnement fourni utilise un serveur NodeJS local avec une connexion WebSocket Socket.IO afin de représenter et d’interagir avec l’environnement. Vous êtes libre d’utiliser le langage et les technologies de votre choix. Vous êtes libre de reproduire l’environnement de la manière que vous le souhaitez si cela facilite/accélère votre développement.

**Objectif du Hackathon**

Fournir une réponse ou un début de réponse à la question suivante :

« Comment feriez-vous pour calculer le chemin optimal à emprunter lors de la procédure de cadenassage ? »

**Livrables**

Pendant la durée de l’événement, vous êtes totalement libres de vos technologies, des librairies utilisées et de tout autre outil qui pourrait vous être utile. L’entièreté du code que vous allez créer pendant cet événement restera en votre possession, mais il sera nécessaire de présenter les résultats que vous avez obtenus avec votre processus pendant la présentation de 10min (chronométrée!) devant le jury (10 minutes pour présenter l’épreuve de web et d’IA).

**Ressources disponibles**

## Informations du jeu

A screenshot of a video game

Description automatically generated

Le jeu comporte une interface graphique affichant la carte ainsi que les contrôles de base de l’agent comme ci-dessus. L’interface graphique permet d’envoyer manuellement des instructions à l’agent en utilisant une connexion WebSocket. Vous êtes invités à explorer le code commenté pour les détails des requêtes.

Le symbole en nœud papillon représente une valve.

Le symbole en bas a gauche représente une machine.

Le statut de la game est représenté par une JSON string d’un tableau en 2-Dimensions. Le statut de la case est 0 si la case est vide, elle est de 1 si l’agent est présent dans la case, elle est de 2 si une valve est présente dans la case, elle est de 4 pour la machine objective. Toute autre valeur est la somme de deux ou plus éléments précédents, et représente que les éléments sont superposés dans la même case.

## Sources externes

1 – Communication WebSocket par socket IO :

<https://socket.io/fr/docs/v4/>

2 – Conversion socket.IO et WebSocket standard:

<https://stackoverflow.com/questions/29604563/websocket-client-to-socket-io>

3 -