



Estimado Comandante Lando Calrissian,

Apreciamos la labor que ha realizado durante todos estos años en Varn y Kessel para la Nueva República. Es notablemente conocida su habilidad en la creación de droides y armamento. Es por ello que necesitamos su ayuda para terminar el nuevo droide de combate YVH al que aún le falta el módulo de selección de objetivos a atacar.

Requisitos de la misión

Los módulos del YVH disponen de un sofisticado sistema de comunicación entre ellos mediante peticiones a un API HTTP. El objetivo de la misión es desarrollar un endpoint HTTP que acepte datos JSON y devuelva datos JSON y una serie de peticiones para poder obtener el historico de calculos realizados para su auditoría.

El módulo de visión enviará una petición **POST** a **/radar** con la información que recibe de su entorno, y el módulo que usted debe desarrollar deberá devolver cuales son las coordenadas del objetivo visible que debe de ser atacado.

Un ejemplo de cuerpo de envío sería:

```
JavaScript
{
  "protocols":["avoid-mech"],
  "scan":[{
    "coordinates":{"x":0,"y":40},
    "enemies":{"type":"soldier","number":10}
  }]
}
```

- **protocols** : Protocolo o lista de protocolos que han de ser usados para determinar cual de los siguientes puntos debe de atacarse primero.
- **scan** : Lista de puntos extraidos del módulo de visión, que es un array de puntos con el número de objetivos de esa posición, y los siguientes subvalores:
 - **coordinates** : Coordenadas x e y del punto.
 - **enemies** : Tipo de enemigo `type` y número `number` . Los posibles valores de `type` serán: `soldier` y `mech`.
 - **allies** (optional) : Número de aliados que hay en dicha posición. Si no está presente este valor, significa que no hay aliados en la zona.

La respuesta debe de contener las coordenadas x e y del siguiente punto a destruir. Un ejemplo de cuerpo de respuesta para el ejemplo anterior sería: `{"x":0,"y":40}`

De esa manera, nuestro droide de combate YVH sabría cual es el siguiente elemento que debe destruir.

Para determinar cual es el siguiente punto a destruir, deben de tenerse en cuenta cuales son los protocolos solicitados, y actuar según sus reglas. Protocolos disponibles:

- **closest-enemies** : Se deberá priorizar el punto más cercano en el que haya enemigos.
- **furthest-enemies** : Se deberá priorizar el punto más lejano en el que haya enemigos.
- **assist-allies** : Deberan de priorizarse los puntos en los que exista algún aliado.
- **avoid-crossfire** : No debe de atacarse ningún punto en el que haya algún aliado.
- **prioritize-mech** : Debe de atacarse un mech si se encuentra. En caso negativo, cualquier otro tipo de objetivo será válido.
- **avoid-mech** : No debe de atacarse ningún enemigo del tipo mech

El módulo de auditoría dispondrá de una serie de peticiones **GET** a `/audit` y `/audit/:id` y **DELETE** a `/audit/:id` donde se podra recibir el historico de calculos realizados

A tener en cuenta

Es importante denotar que podrán proporcionarse **varios protocolos** en la petición. A modo de ejemplo, si recibiésemos los protocolos closest-enemies y assist-allies, deberíamos buscar el punto más cercano que tuviese aliados presentes.

En todo caso se proporcionarán **protocolos compatibles** entre sí. Puede asumirse que en ningún caso el módulo recibirá, por ejemplo, los protocolos closest-enemies y furthest-enemies en la misma petición.

Finalmente es importante tener en cuenta que los objetivos a una distancia superior a **100m** se consideran demasiado alejados para ser atacados y por lo tanto deben ser **ignorados**.

Lo ideal sería que, cuando nuestras fuerzas de inteligencia decidan implementar nuevos protocolos teniendo en cuenta otros factores, la solución propuesta pueda ser extendida con facilidad para aceptar estos nuevos criterios.

Al ser un sistema crítico debe almacenarse en una base de datos persistente las peticiones y resultados obtenidos pudiéndose listar, obtener el detalle o borrar estos registros mediante peticiones HTTP

Instrucciones de entrega

Dado que esta labor es de vital importancia para nosotros, la Nueva República, hemos provisto una serie de casos de prueba que verificarán que el algoritmo funciona correctamente.

Deberás tener curl instalado y ejecutar el comando `./tests.sh` en tu máquina que podrás obtener en <https://static.applivery.com/assets/codetests/backend-codetest-tests.zip>

Se deberán proporcionar una serie de scripts que permitan obtener el listado de las últimas interacciones del módulo, el detalle de una en concreto o borrar alguno de los registros

También se proporcionara un pequeño manual con las instrucciones para su ejecución.

Al finalizar la misión, se precisa que comprima todos los ficheros fuente incluido en un **fichero comprimido** y lo envíe de vuelta por correo electrónico o proporcione acceso a algún repositorio público.

Buena suerte, may the Force be with you.

Requerimientos tecnológicos

Como ingeniero de la Nueva República debes enfrentar este desafío técnico con los siguientes requisitos precisos:

- HTTP framework: Express
- Lenguaje: Typescript
- Base de datos: Mongo

