

Hackathon UAM 2019

Introducción

Introducción

¿Para qué hemos venido?

- Torneo de programación de *IA*
- Combate de tanques
- *JavaScript*

Introducción

¿Qué tenemos que hacer?

- Utilizar el *entorno de desarrollo*
- Crear un script que *pilote* un tanque
- Luchar con otros robots en la *arena*

Introducción

¿Qué necesitamos?

- Un *ordenador*
- Un navegador reciente
 - recomendamos: *Google Chrome*
- Un editor de texto
 - *Atom, Sublime, Notepad++, ...*

Introducción

¿Cómo entrego mi código?

- Por email a: *elias@redradix.com*
- Asunto: *[Hackathon UAM] nombre-de-tu-equipo*
- Adjunto: *robot.js*

Entorno de Desarrollo

Entorno de Desarrollo

Para empezar:

- Descarga el *zip con el entorno de desarrollo*
 - <https://hackathon.redradix.com/uam>
- Abre *index.html* con *Chrome*
- Abre *robot.js* con un editor de texto

Entorno de Desarrollo

Pantalla de selección

- En la *arena* pueden luchar hasta *6 tanques*
- Tienes unos cuantos *bots* para practicar
- Tu robot aparece como: *yours*
- *Configura* la arena a tu gusto y *observa* el combate

Entorno de Desarrollo

Bots incluidos:

- *dummy*: blanco inmóvil
- *rabbit*: se mueve mucho pero no dispara
- *rook*: dificultad baja
- *counter*: dificultad media
- *grunt*: dificultad alta

Control del Tanque

Control del Tanque

robot.js

- Declara una función *main*
 - punto de entrada del programa
- Recibe *un parámetro*
 - *tank*: nuestro tanque robot
- Controlamos el tanque llamando a métodos de *tank*

```
async function main(tank) {  
    // tu código aquí  
}
```

Control del Tanque

La arena es un *rectángulo* de **1340x1000**

- El origen **(0, 0)** es la *esquina inferior izquierda*
- El ángulo **0°** es el *Este* y el ángulo **90°** es el *Norte*

Control del Tanque

Cuatro métodos para consultar el estado del tanque

- *await* **tank.getX()**: coordenada X
- *await* **tank.getY()**: coordenada Y
- *await* **tank.getSpeed()**: velocidad real del tanque
- *await* **tank.getDamage()**: daño acumulado

```
async function main(tank) {  
  console.log(await tank.getX());      // 461  
  console.log(await tank.getY());      // 789  
  console.log(await tank.getSpeed());  // 0  
  console.log(await tank.getDamage()); // 0  
}
```


Control del Tanque

Tres métodos para realizar acciones

- *await* **tank.drive(angle, speed)**: avanzar
- *await* **tank.shoot(angle, range)**: disparar
- *await* **tank.scan(angle, resolution)**: escanear

Control del Tanque

`tank.drive(angle, speed)`

- *angle*: ángulo en el que queremos avanzar
- *speed*: (max 100) potencia del motor
- *retorno*: velocidad real del tanque
- el motor sigue en marcha hasta que lo paremos

```
async function main(tank) {  
    await tank.drive(0, 75);  
    while (await tank.getX() < 800) {  
        null;  
    }  
    await tank.drive(0, 0);  
}
```

Control del Tanque

tank.scan(angle, resolution)

- *angle*: ángulo en el que queremos escanear
- *resolution*: (max 10) apertura del escaner
- *retorno*:
 - si hay un tanque: distancia al tanque
 - si no hay tanque: 0

```
async function main(tank) {  
    console.log(await tank.scan(0, 10)); // 0  
}
```

Control del Tanque

`tank.shoot(angle, range)`

- *angle*: ángulo en el que queremos disparar
- *range*: (max 700) distancia a la que explotar
- *retorno*: *1* o *2* si se ha disparado; *0* en caso contrario.
- solo puede haber *dos* misiles en el aire

```
async function main(tank) {  
    await tank.shoot(0, 200);  
    await tank.shoot(0, 200);  
}
```

Control del Tanque

Consejos generales

- La funcion *main* nunca deberia terminar
 - envuelve tu código en un *while (true) { ... }*
- Recuerda poner *siempre* **await** al llamar a un método
- ¡Cuidado con los bucles infinitos!
- El tanque tiene inercia. Frena antes de llegar.

Control del Tanque

Consejos generales

- La funcion *main* nunca deberia terminar
 - envuelve tu código en un *while (true) { ... }*
- Recuerda poner *siempre* **await** al llamar a un método
- ¡Cuidado con los bucles infinitos!
- El tanque tiene inercia. Frena antes de llegar.

Reglas del Juego

Reglas del Juego

- Un tanque es destruido al acumular 100 puntos de daño
- Impactar con la pared: 5 puntos de daño
- Impactar con otro tanque: 5 puntos de daño (a los dos)
- Misiles: 1-25 puntos de daño (según distancia explosión)
- Último tanque en pie gana
- En caso de tablas: el combate se finalizará manualmente