



# Tank AI

**Ion Laurentiu Cristian**  
International Computer Highschool of Bucharest

## Descriere

**TankAI** este o platforma destinata cercetarii si educatiei in Inteligenta Artificiala. Inspirat din RobotWar

( <https://en.wikipedia.org/wiki/RobotWar> ) "The task set before you is: to program a robot, that no other robot can destroy!"

TankAI ofera o interfata de programare (API) pentru a experimenta cu diverse metode (Machine Learning, Brute-Force Search, Heuristic Search, Q-Learning, Neuroevolution, Symbolic Planning, etc) si a crea cel mai inteligent agent artificial care sa se descurce in orice situatie.

Tancurile nu au acces la informatii despre ceilalti "jucatori", si se lupta (autonom) in arena pana cand ramane numai unul. Se pot crea turnee pentru a determina care este cel mai bun dintr-o serie de runde, si astfel programele pot sa adune date si sa se antreneze pentru a deveni mai inteligente.

## Pornirea aplicatiei

```
python main.py
```

## Optiuni rulare

```
-T      mod testare
-t      turneu
-n      nr. de batalii din turneu
-g      mod fara grafica
-D      resetare baza de date
```

## Configurare

Valorile implicite sunt stocate in defaults.py  
Pentru a schimba aceste valori adaugati in conf.py

## **Crearea de tancuri noi**

Fisierul template.py contine un sablon de la care puteti porni.

Adaugati numele tancului in conf.py

```
ex.      t='untanc'  
         tanks.append(t)
```

In noul program creat exista 2 functii:

### **initialize()**

Aici se adauga codul de start, nu are niciun parametru, si trebuie sa returneze in mai putin de o secunda, altfel tancul este eliminat.

### **respond()**

Aici se adauga codul prin care tancul raspunde la mediul inconjurator.

Este apelata de 60 de ori pe secunda si trebuie sa returneze in mai putin de 0.015 s, altfel tancul este eliminat.

## **Senzori**

Senzori disponibili:

Utilizare: variabila = self.sensors['SENZOR']

**TICK** – timpul de cand a pornit tancul.

**HEALTH** – 0-100

**POS** – pozitia fata de centrul arenei – (x,y)

**TUR** – unghiul tunului fata de tanc

**PING** – rezultatul trimiterii unei raze

returneaza un tuple cu 3 valori  
**tipul:** 'w' wall, 't' tank sau 'b'

bullet

**unghiul** tunului cand a fost trimis

ping-ul

**distanța**

**GYRO** – unghiul tancului fata de arena

**HEAT** – cat de mult s-a supraincalzit tunul  
o data ce tunul ajunge la o anumita  
valoare nu mai poate trage pana ce se raceste  
exista si posibilitatea ca incercand  
sa traga in timp ce e supraincalzit, tunul sa se  
blocheze, astfel primind o pedeapsa in timp pana  
cand poate sa traga din nou

**LOADING** – reincarcarea tunului dureaza. Acest  
senzor spune cat de mult mai este pana cand tunul  
este gata sa traga din nou

exista si posibilitatea ca  
incercand sa traga in timp ce e incarcat, tunul sa  
se blocheze, astfel primind o pedeapsa in timp  
pana cand poate sa traga din nou

## **Actiuni**

Tancul poate actiona:

**force()** - acceleratie

self.force(100) #viteza maxima in fata

self.force(-50) #jumătate din viteza in marsalier

**torque()** - directie

self.torque(100) # intoarce in sensul acelor de  
ceasornic cu viteza maxima

self.torque(-50) # intoarce in sensul invers  
acelor de ceasornic cu jumătate din viteza

**turret()** - rotirea tunului

**fire()** - trage

**self.fire()** # lanseaza proiectilul cat se poate de departe

**self.fire(10)** # explodeaza dupa 10 metri

**ping()** - trimite un puls radar in directia tunului  
la urmatorul TICK se poate citi senzorul  
'PING' pentru rezultate

**log()** - mesaj in fisierul jurnal al tancului  
folositor pentru a depana sau antrena  
programul

