Intelligent Systems 2014

Planet Wars Bots



**Authors**

Anouk Boukema

Baudouin Duthoit

Gossa Lô

# Introduction

Blablabla

# Bot #1 : FirstBot

## Pseudo code :

DoTurn():

Source = FindThePlanetWithTheBiggestFleetWeOwn();

PossiblesDestinations = InterestingPlanets();

Destination = FindTheFirstCapturablePlanetOf(PossiblesDestinations);

IssueOrder(Source,Destination);

InterestingPlanets():

List;

for (planet in NotMyPlanets):

If (planet.GrowthRate > planet.Fleet):

List += planet;

List += enemyPlanets;

return List;

## Explanations :

This first code we implemented find the biggest fleet available by scrolling through all our planets to find the one with the biggest fleet. This method will be used for most of the algorithms.

To find the destination, we first sort all the available destinations. To do so, we first choose all planets with more GrowthRate than Ships on it (we'll get paid back thus), then we'll add all the ennemy planets, because it's always beneficial to attack the enemy. Then, with this sorted list of planets, we choose the first one that we can capture. This bot is espacially efficient in serial mode.

## Why ? :

# Bot #2 : HillclimbingBot

## Pseudo code :

DoTurn():

source, dest, D=-100;

For (s in possibleSources)

For (d in possibleDestinations)

If (DCalculation(s,d)>D)

source = s;

dest = d;

D = DCalculation(s,d);

IssueOrder(source,destination);

DCalculation(source,dest):

return MyShipsLoss - EnemyshipsLoss + MyGrowthRateImprove - HisGrowthRateImprove

## Explanations :

This code is just finding the best attack to do depending on our heuristic function : DCalculation(). For this, we just go through all couple of source/destination and calculate the euristic value of it.

In DCalculation, we only count the difference between the current state and the next one, by comparing, the loss of ships and if the growthrate is increasing for us and/or decreasing for the enemy.

## Why ? :