

File:

```
10
bike, 1, 5
tv, 4, 4
car, 2, 1
computer, 5, 3
motorbike, 3, 2
```

1st Iteration:

My first Iteration for a negotiator robot was to simply have the robot always send back the best possible offer he could make to the random negotiator. Now while this is a very ineffective method for negotiating it will either always result in the most possible utility or a failed negotiation causing (in my case) both negotiators to lose -5 utility. I simply wanted to see if being super greedy in your offers would result in better overall points, and it does but not nearly enough to make up for all the failed negotiations you have to make in the process.

Outcome (only two are possible), ***NOTE* I am Negotiator B:**

<u>My offer (every time):</u> [bike, tv, computer, motorbike, car]	<u>My offer (every time):</u> [bike, tv, computer, motorbike, car]
<u>Their Offer: (random)</u>	<u>Their Offer: (random)</u>
Failed negotiation: Negotiator A: -5 Negotiator B: -5	Successful negotiation: Negotiator A: 2 Negotiator B: 10
Final result: Negotiator A: -5 Negotiator B: -5	Final result: Negotiator A: 2 Negotiator B: 10

2nd Iteration:

My second Iteration for a negotiator robot was to have the robot compare the offer in regards to its own utility versus the other negotiator's utility and if my own utility is higher utility I accept the offer thus always getting a higher score. For this iteration I still kept my offer the same, trying to get the best possible offer I can each time.

Outcome, ***NOTE* I am Negotiator B:**

<u>My offer (every time):</u> [bike, tv, computer, motorbike, car]	<u>My offer (every time):</u> [bike, tv, computer, motorbike, car]
-----------------------------------------------------------------------	-----------------------------------------------------------------------

<u>Their Offer: (random)</u> Round 1: A:[motorbike, bike, computer, tv, car] B: None Successful negotiation: Negotiator A: 0 Negotiator B: 10 Final result: Negotiator A: 0 Negotiator B: 10	<u>Their Offer: (random)</u> Round 1: A:[motorbike, tv, computer, car, bike] B: None Round 2: A:[car, motorbike, bike, computer, tv] B:[bike, tv, computer, motorbike, car] Successful negotiation: Negotiator A: -2 Negotiator B: 10 Final result: Negotiator A: -2 Negotiator B: 10
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3rd Iteration:

My third Iteration I decided to add a lot more features than the previous versions in an attempt to throw off the other negotiators. So instead of trying to devise a clever way to figure out how the other robots are working/negotiating I decided to randomly generate offers (in order to make it more difficult for the negotiators to try to figure out what I am doing) that would generate me more utility than the last offer given to me. Also on the first offer (for negotiator A or B) I always offer my best offer simply because there is no reason not to. As for the final offer (assuming I am negotiator B) I would offer my best possible outcome again. If I am negotiator A and I get to round 10 I either accept the offer (if it generates me more utility) or decline the offer (if my opponent gets more utility, sort of a “if I can’t win neither can you” mentality). In addition I always accept offers where the other robot has a negative utility and I have a positive utility or where my utility is higher than theirs for the offer (Sometimes the offers are equivalent and it accepts but I assume this is due to my code rounding because it is compiled in python 2.x and not 3.x but this should fix itself when it is compiled in the right version). Also to note because of the way I implemented my robot and its negotiations it will ALWAYS tie (both will lose) when it is versus itself.

Outcome, ***NOTE* I am Negotiator B:**

TRIAL 1: Round 0: A:[computer, tv, motorbike, bike, car] B: None Round 1:	TRIAL 2: Round 0: A:[bike, computer, motorbike, car, tv] B: None Round 1:
----------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------

<p>A:[motorbike, tv, car, computer, bike] B:[bike, tv, computer, motorbike, car] Round 2: A:[bike, motorbike, tv, car, computer] B:[bike, tv, computer, motorbike, car] Successful negotiation: Negotiator A: 0 Negotiator B: 4 Final result: Negotiator A: 0 Negotiator B: 4</p> <p>TRIAL 3: Round 0: A:[motorbike, computer, car, tv, bike] B: None Round 1: A:[car, computer, motorbike, tv, bike] B:[bike, tv, computer, motorbike, car] Round 2: A:[motorbike, tv, bike, computer, car] B:[bike, tv, computer, motorbike, car] Successful negotiation: Negotiator A: 2 Negotiator B: 10 Final result: Negotiator A: 0 Negotiator B: 10</p>	<p>A:[tv, computer, bike, car, motorbike] B:[bike, tv, computer, motorbike, car] Round 2: A:[tv, computer, motorbike, bike, car] B:[bike, tv, computer, motorbike, car] Round 3: A:[bike, motorbike, computer, car, tv] B:[computer, tv, bike, motorbike, car] Successful negotiation: Negotiator A: 0 Negotiator B: 4 Final result: Negotiator A: 0 Negotiator B: 4</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

File 2:

```
10
bike,1,5
tv,4,4
car,2,1
computer,5,3
motorbike,3,2
watch,7,6
helmet,6,7
```

Outcome, *NOTE* I am Negotiator B:

<p>TRIAL 1: Round 0: A:[motorbike, tv, computer, car, bike, helmet, watch] B: None</p>	<p>TRIAL 2: Round 0: A:[helmet, motorbike, watch, bike, tv, car, computer] B: None</p>
--------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------

<p>Round 1: A:[car, computer, motorbike, helmet, watch, bike, tv] B:[helment, watch, bike, tv, computer, motorbike, car] Round 2: A:[tv, bike, computer, helmet, motorbike, watch, car] B:[tv, bike, motorbike, computer, helmet, car, watch] Successful negotiation: Negotiator A: 0 Negotiator B: 2 Final result: Negotiator A: 0 Negotiator B: 2</p> <p>TRIAL 3: Round 0: A:[computer, car, bike, tv, helmet, motorbike, watch] B: None Round 1: A:[helmet, motorbike, tv, car, computer, watch, bike] B:[helmet, watch, bike, tv, computer, motorbike, car] Round 2: A:[bike, tv, watch, motorbike, car, computer, helmet] B:[helmet, watch, bike, tv, computer, motorbike, car] Successful negotiation: Negotiator A: -4 Negotiator B: 0 Final result: Negotiator A: -4 Negotiator B: 0</p>	<p>Round 1: A:[watch, car, motorbike, tv, computer, bike, helmet] B:[helmet, watch, bike, tv, computer, motorbike, car] Round 2: A:[bike, helmet, watch, car, tv, motorbike, computer] B:[helmet, tv, computer, watch, car, bike, motorbike] Successful negotiation: Negotiator A: 0 Negotiator B: 6 Final result: Negotiator A: 0 Negotiator B: 6</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Conclusions:

In conclusion my robot negotiator is very effective at maximizing on the faults of other robots and minimizing their chance for points, in that if I am going to lose I make sure we both lose taking the hit just to ensure that they don't win the negotiation. Since I included a random factor inside of my robot it makes it very difficult for other robots to determine what I am doing thus

giving the illusion of my robot mindlessly doing things when in fact it waits for the other negotiator to make a mistake and then capitalize on it. I also determined the final implementation of my robot cannot argue with itself and always results in a failed negotiation.