Project assignment: prisoner's dilemma

The **prisoner's dilemma** is an example of a game analysed in the field of game theory.

Why Dilemma:

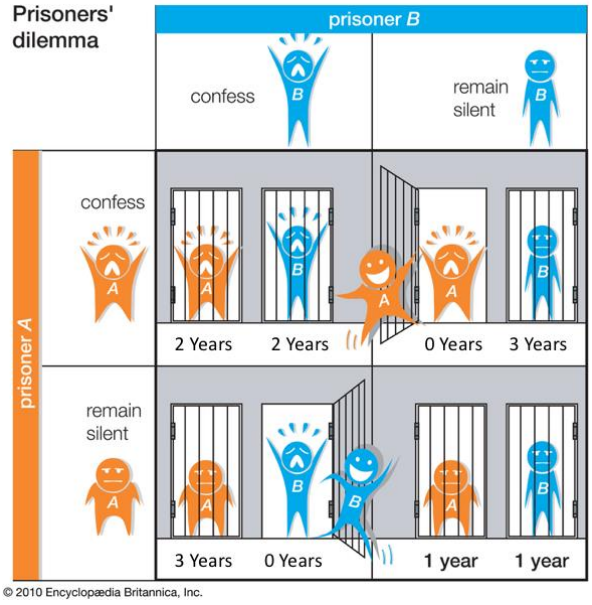
Because the game shows how two completely rational individuals might not cooperate, even if it appears that it is in their best interests to do so.

Our Goal:

We are running a set of experiments that may help us to draw initial conclusions of human behaviour in the context of decision-making processes.
In this particular game, we focus on the decisions humans take when they are faced with two choices. Nevertheless, their win depends not only on their choice but on the choice of their opponent.
The output of this experiment will allow us to build a basic understanding of how humans behave in such situations and how their behaviours adapt over time. Therefore, please when you are done with the implementation and the test of your Game, **PLAY THE GAME HONESTLY.**

Game description:

Two members of a criminal gang are arrested and imprisoned. Each prisoner has no means of communicating with the other. The prosecutors lack sufficient evidence to convict the pair on the principal charge, but they have enough to convict both on a lesser charge. Simultaneously, the prosecutors offer each prisoner a bargain. Each prisoner is given the opportunity either to betray the other by testifying that the other committed the crime, or to cooperate with the other by remaining silent. Here is the offer is:



- If A and B each betray the other, each of them serves two years in prison
- If A betrays B but B remains silent, A will be set free and B will serve three years in prison (and vice versa)
- If A and B both remain silent, both of them will only serve one year in prison (on the lesser charge).

Tasks:

- Develop a web application that allows to play the game **prisoner's dilemma**. There are two ways to play the game: against another player or against the computer. Both ways should be implemented.
- One game consists of **30 rounds**. In each round both players give their choice (remain silent/confess). After each round they get the conviction: x years in prison.
- Each game is saved in a **database**. The information to be stored is: a unique id for the game, the name of both players, their 30 choices and convictions in order.
- Use **web sockets** for the communication during the game between the webserver and the browser.
- Provide a **REST web service**. This service can be used to collect the results of all games played or to collect the games from a specific player. The **contract** of the REST service is describe in the file gamesSwagger.json. You can open this file in the online-editor http://editor.swagger.io/
- Use the **data layer** PrisonerDilemma.jar. This library implements the functionality for the game: virtual players, determination of the conviction, … The project ExampleUsePD demonstrates its functionality. The file javadoc.zip contains the Javadoc for the data layer.
- Use the github **repository** https://github.ugent.be/gdtProject2018/groep-XX, where XX is your group number. Every member of the team pushes his/her code under his/her name. Post regularly to your repository.
- Make sure your project can be tested with Docker! In your repository there should be a map **docker** which contains a file **project.yml** and all necessary information to run your docker container(s) with the following command:

```
docker-compose -f project.yml up --build
```

- **Deploy** the web application and the webservice online.
- Programming languages, frameworks and cloud environment are free to choose.

Submitting:

- The deadline for this project is 22 december 2018 at noon (12:00 hrs)
- All your code should be in your github respository
- All configuration for docker should be in the map docker in your github
- Your web application and REST service should be deployed online
- Both team member should have played a game (=30 rounds) against a computer player and against each other. This game data should be provided by the online REST service. **PLAY THE GAME HONESTLY!**
- Your project will be tested online and locally!
- All information necessary to test the project is in the readme.md on github (f.i. the url of the online deployment, how to use the web application, …)