

Technical instructions for submitting the Python program

Go to <http://www.python.org/download/>, download the Python 3.3.2 installer and execute. (This website also contains useful documentation of the Python programming language.)

In the course webpage <http://www.wisdom.weizmann.ac.il/~feige/agt2013.html> there will be the following files (place them all in the same directory).

The file *StudentName.py* is an example of an implementation of an agent, the class *AgentStudentName* (*class* is a technical term in Python). This serves as a template that you can modify in order to create your own agent.

The file *Definitions.py* contains some classes needed for *StudentName.py*.

Given your own version of the file *StudentName.py* with *AgentStudentName* class (e.g., after you have modified it to implement your own agent), the file *Check.py* can be used in order to check whether your agent is valid. Run *Check.py* (in IDLE, which is a Python editor that comes with the installation, press 'F5'), to make sure your agent is valid - only valid agents can play.

Only after you are satisfied that your agent runs properly, change the name of the file *StudentName.py* to your own (true) name (e.g., *RoeDavid.py*) or (true) names if submitted in pairs (e.g., *AliceCohenBobLevi.py*). Likewise, change the name of the class, replacing *StudentName* by the name of the file (e.g., to *AgentRoeDavid*). Submit only your renamed file.

The Agent class

The agent *class* has four *methods* (see the code example for how to use these methods).

1. *__init__(self, IdNumber, NumberOfPlayers, NumberOfRounds)*. This method informs the agent its id number, the total number of players (the players are 0, 1, ..., $\text{NumberOfPlayers} - 1$), and the total number of rounds (which is intended to be 100 - if this will change, you will be notified before you submit your project).
2. *RecievedGame(self, i)*. This method informs the agent that the agent with id *i* invites him to a game, and returns a *Move* object with the desired move.
3. *InitiateGame(self)*. This method requests the agent to invite another agent to a game. The method returns an *Invitation* object with the id of the other agent and the desired move.
4. *SecPlayerMoveLastGame(self, Result)*. This method gives the agent a *Move* object with the move played by the other player in the match.