

Design a Game Player

Name: _____

Choose one of the iterated game tournaments you want to design a player for (Check the box)

Prisoner's Dilemma ☐

0 = Stay Silent

1 = Betray Partner

Chicken ☐

0 = Swerve

1 = Continue

Bach or Stravinski ☐

0 = Bach

1 = Stravinski

If you have extra time this week, you can try making players for the other two games you did not choose!

Your player:

- Includes its own name
- Has one function called `next_move()` that **returns** the move it wants to make next (0 or 1)
- Has access to its own move history in a list called "my_history"
- Has access to its opponent's move history in a list called "other_history"
- Can include other functions to be used by `next_move()`

The Tournament:

- Your player will face off against all other players of the same game
- In each face-off, the players will play the game 200 times
- The player wins the match if it gets the same or higher score as its opponent after 200 games
- The player with the highest number of wins wins the tournament!

Name of your Game Player:

Strategy Requirements:

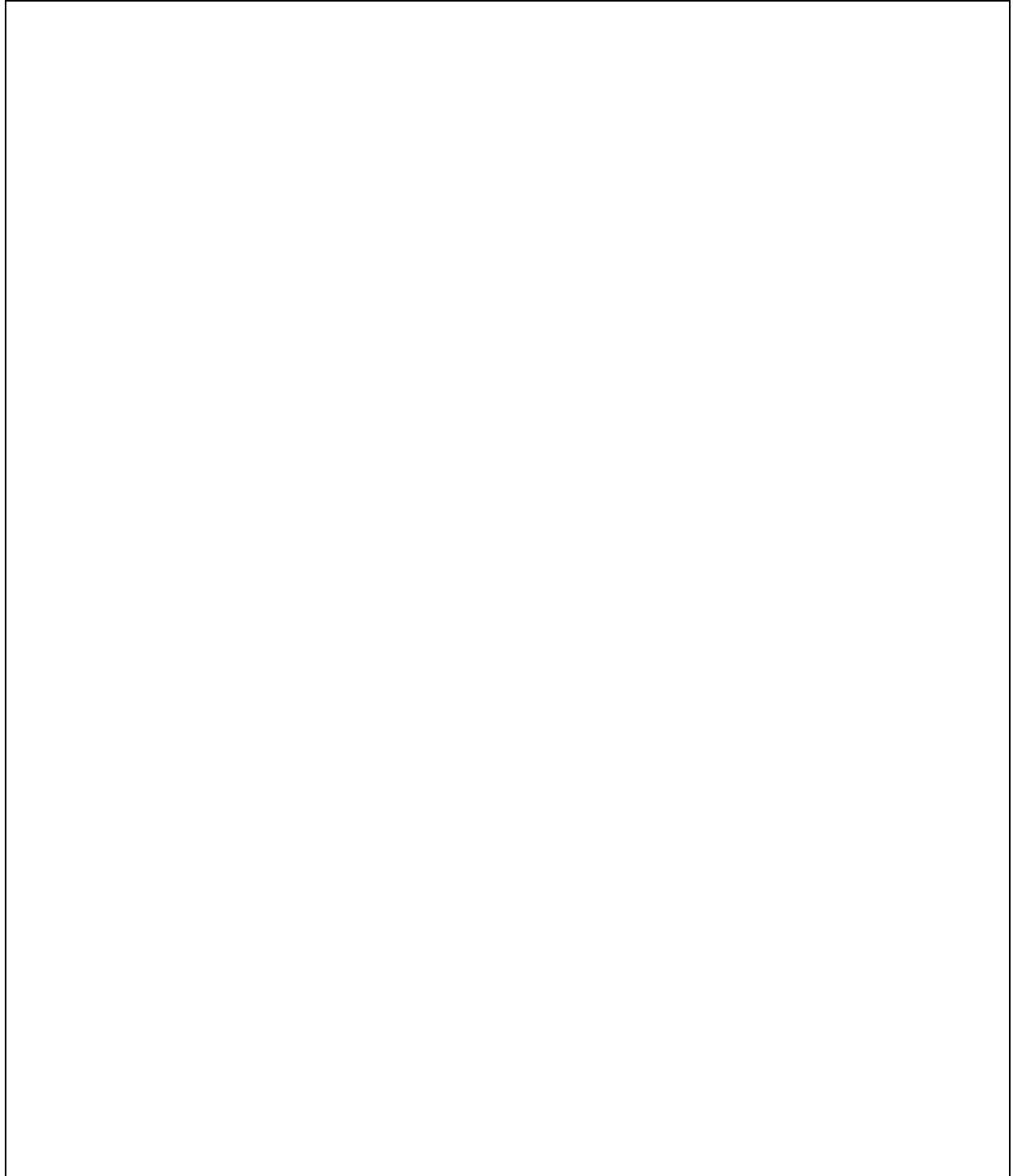
- Your game player cannot give the same answer in all circumstances
- Your game player must include at least one "if" statement
- Apart from the above restrictions, you may make your strategy any way you want!

There will be a template program and testing software available!

You must turn in this document before you begin programming!

Create a diagram of your decision-making process here!

It can be a flowchart, pseudocode, or any other format that clearly shows what the player will do when asked for a decision.

A large, empty rectangular box with a thin black border, intended for the student to draw a diagram of their decision-making process. The box is currently blank.