**Pirate Bot Game Instructions**

by Saar Nehemia

Table of Contents

[**Pirate Bot Game Instructions** 1](#_Toc169277701)

[**Introduction** 2](#_Toc169277702)

[**Entities** 2](#_Toc169277703)

[Board 2](#_Toc169277704)

[Ship 2](#_Toc169277705)

[Island 3](#_Toc169277706)

[Player 3](#_Toc169277707)

[Block 3](#_Toc169277708)

[**Developing Bots** 4](#_Toc169277709)

[API 4](#_Toc169277710)

[Your code 4](#_Toc169277711)

[**Checking Your Bot Performance** 5](#_Toc169277712)

[**Versions** 6](#_Toc169277713)

[Version 1.0 6](#_Toc169277714)

[Version 2.0 6](#_Toc169277715)

[Next versions 6](#_Toc169277716)

## **Introduction**

The pirate bot game is a turn-based game. In each run a series of games will run to check which player develops the most robust strategy.

* Currently, all games are with the same board and the same maximum number of turns allowed.
* In the beginning of each game, we randomly choose the order of the player turns.
* In each turn, the current player code is running (only one player plays).
* In each turn, each player’s ship can move once so you can command your fleet on parallel missions.
* If the code crashes, the player loses.
* After each turn, we check if any player achieved the victory criterion. If so, game ends.
* If maximum number of turns reaches the game end and the result is draw.
* In the end of all games, we print the total results: number of wins for each player and number of draws.

## **Entities**

### Board

The game board in which your bots will compete. You can see all available board under **boards**.

Each board contain the following settings:

* Board\_size: notice it can change from one board to another.
* Blocks: list containing all blocks objects. Location units is relative to board\_size.
* Islands: list containing all islands objects. Location units is relative to board\_size.
* Player\_base\_island\_indices: assign a base island for each player. The position in the list represents which player gets this index and the value is the index in islands list defined above that is assigned to this player. For example: Player N gets the Nth index and islands[Nth index] is assigned to be his base island. In the example below: player 0 gets island[1] and player[1] gets island[0].
* Players\_ship\_speed: each player can have different ship speed. same explanation as above for the notation. Units are relative to board\_size
* Players\_num\_ships: each player can have different speed. same explanation as above for the notation. Units are relative to board\_size.
* Victory criterion: right now, there is only one option – num islands to conquer.

### Ship

Ships are your soldiers, they can:

* Move across the board
* Capture islands
* Increase/decrease islands life (dock on island)
* Destroy other ships (whether in sea or on island).

Each ship has the following attributes:

* ship id
* player id
* ship speed
* location
* is\_moved: True/False indicator if ship already moved in this turn.
* frontend\_obj: object to draw that represents ship on the screen.

General rules:

* Ship staying on your island increases your island life by 1 each turn.
* Ship entering neutral island capture it on the same turn.
* Ship entering empty enemy island decreases island life by 1 each turn. If there are enemy ships on the island the ship kills one enemy ship and itself.
* Once captured island life reaches zero, it becomes neutral again.
* If two ships collide in the sea, they are both destroyed (whether they are of the same player or not).

### Island

Islands are places on board you want to capture to win the game.

Each island has the following attributes:

* Island id
* Own player id: equals -1 if island is neutral (aka no player owned the island).
* Location
* Ships: list of all ships that are docking in island right now
* Current life: HP of island.
* Ship\_creation\_time: number of turns after which island will generate ship of owned player.
* Frontend\_obj: object to draw that represents island on the screen.

### Player

Players are the objects which hold all yours and your enemies information.

Each player has the following attributes:

* Player id
* Player name
* Player class: stores the bot you will write.
* Ships
* Max\_ship\_id: to create a new ship with new id that has not been assigned before.

### Block

Blocks are locations on board you cannot move through.

Each block has the following attributes:

* block id
* location
* current life (currently not used): in future versions, ships will be able to destroy blocks.

## **Developing Bots**

### API

Access to current game and board status (players, blocks, islands and ships) is available through the API. You can see all available functions + documentation in classes\api.py.

**Don’t access any API’s attributes or underscore methods – they exist for internal use only! Using them will allow you to illegally change the game (will be solved in future versions).**

Let me know if you think more functions should be added to the API or if you find bugs/hucks.

### Your code

* You need to create python code in “**players**” folder with your name.
* **You don’t need to touch or change any other code other than this. Feel free to look at it if necessary.**
* In this code you must put all your strategy for a general turn in the following function:

def do\_turn(game\_api: game\_api.API):

* You can access all available API functions easily using game\_api.{relevant\_method}.

## **Checking Your Bot Performance**

You can use the main script ‘main.py’ to run a series of games to check your code strategy performance.

In ‘main.py’ you can define:

1. player\_names: who plays against who.
2. Board\_name: on which board the games take place.
3. Num\_games: how many games you wish to run.
4. Max\_num\_turns: game timeout.
5. To\_draw\_game: whether you want to see the graphics of each game or only the results.
6. debug\_mode: you can run in debug mode to debug your code if it crashes.

After running ‘main.py’, you will see prints of each game progress and in the end the count of wins for each player and counts of draws.

The script ‘main.py’ set the above parameters and runs a series of games through the class ‘game.py’ which runs a single game.

**Good luck and have fun - may the best pirate win!**

## **Versions**

### Version 1.0

**Release date:** 13.05.24 – Independence Day!

**Features:** written above.

### Version 2.0

**Release date**: 14.06.24.

**Features added:**

* **Code setup:**
  + add requirements.txt to easy setup of game on your computer.
  + Update ReadMe and instructions docs.
* **Ships movement**
  + Now, **all** player ships can move **once** in player turn (in previous version you can only move one ship each turn).
  + If you try to move ship twice in the same turn, your will lose the game.
* **Ships creation**
  + Each island has a timer for creating ships for the player owning them.
  + After each turn, we check for each island if the timer reached zero.
  + If it has no ships or only friendly ships – we create a new ship.
  + If it has enemy ships – we destroy the first enemy ship on the island since a new ship was “created” for the player owning the island and it collided with the enemy ship.
* **Game progress on screen**
  + Players status: color, num islands, num ships
  + Islands status: life, num ships (color according to player or black if neutral)
  + Game turn: current turn out of max turns.
  + Result: prints who won (or draw) at the end of each game.
* **Debug mode**
  + Now you can choose to run a series of games in debug mode and debug your code if an exception is raised.
  + Previously the program told you your code crashed and continued to the next game immediately.
  + Set debug\_mode Boolean parameter in main.py as you wish.

### Next versions

* **New rules:**
  + Different kinds of ships (boat, man o’ war, etc.…) with different stats: ship life, ship strength, ship speed.
  + New option for players: exchange island life for ship creation. Better ship costs more life.
  + Allow to move **up to** maximum speed of ship and not **only in it.**
  + Ships will be able to destroy blocks and allow for ships passage.
  + Add different criteria for victory: destroy all or X enemy ships, capture all or X islands, capture enemy base island, etc. Can add combinations of all.
  + Whirlpools / sea monsters – tiles that kills ships.
  + Add an option to play in simultaneous mode: in each turn all players play simultaneously and thus we can achieve a draw.
* **Gameplay additions**
  + Add more boards and bots.
  + Add some basic strategies to the API which you can use.
  + Add option for more than 2 players.
* **Visual improvements**
  + Island with ships looks different.
  + How to nicely visualize ship collided with blocks.
* **Any more suggestions?**