Macro and Micro Manual

> Sioux Weekend of Code 2017

Macro

> Control your UFOs to conquer planets and gain credits.

Conquering Planets

The goal of Macro is to command your UFOs to move to planets and conquering them.
You can conquer planets that have already been conquered by other players.
If you try to conquer a planet while one or more enemy UFOs are in nearby the planet, you will have to battle (Micro) for the right to conquer the planet.
The winner of the battle will conquer the planet and in case of a draw, no one will conquer the planet.

See commands: conquer, moveToPlanet, and moveToCoord.

Credits

When a planet is conquered you gain credits. These are used to buy more UFOs. The player with the most credits at the end of the game will win so be mindful of how you are spending your credits.

The base income is a **500** credits per second. On top of that you get an additional **50** credits per planet you have conquered per second.

See commands: buy

Receiving Game State from Macro

On each tick of the game, Macro will send the **game state** to all Macro scripts. Macro will wait for **250 milliseconds** during each tick.

Example game state:

```
{
    "Id": 1337,
    "Name": "",
    "Tick": 0,
    "SolarSystems": [
        {
            "Id": 1,
            "Name": "SolarSystem1",
            "Coords": {},
            "Planets": [
                {
                    "Id": 1,
                    "Name": "Planet1",
                    "OrbitDistance": 10,
                    "OrbitRotation": 10,
                    "OwnedBy": 1
                }
            ]
```

```
            }
        ],
        "Players": [
            {
                "Id": 1,
                "Name": "Player1",
                "Credits": 9000,
                "Ufos": [
                    {
                        "Id": 1,
                        "InFight": true,
                        "Coord": {
                            "X": 25,
                            "Y": 75
                        }
                    }
                ]
            }
        ]
    }
```

Sending Commands to Macro

You can send commands to Macro by writting them to `stdout`.
There are 4 types of Macro commands:

- conquer
- moveToPlanet
- moveToCoord
- buy

Macro will wait for **250 milliseconds** (a tick) before processing all commands. Within that time you can send as many commands as you like and they will be queued in Macro for processing. At the end of the tick, Macro will process all the commands

conquer

In order to conquer a planet you have to have at least one UFO that is within a radius of **256** "space meters" of the planet that you want to conquer, and you have to send the `conquer` command with the ID of that planet.

Example: lets say you have UFO at (45,75) and you want to conquer planet 42 which is located at (50,60). The distance is sqrt((50-45)(50-45) + (60-75)(60-75)) = 15 which is within the radius of **256**.

Sending the following `conquer` command will let you conquer planet 42.

```
    {
        "Command": "conquer",
        "PlanetId": 42,
    }
```

moveToPlanet

To move your UFO(s) towards a planet you can send the `moveToPlanet` command with the list of UFOs that you want to move and the ID of the planet you want to move to.

Sending the following `moveToPlanet` command will start moving UFOs 1, 2, and 3

towards planet 42.

```
{
    "Command": "moveToPlanet",
    "Ufos": [1, 2, 3],
    "PlanetId": 42
}
```

moveToCoord

To move your UFO(s) towards a specific coordinate you can send the `moveToCoord` command with the list of UFOs that you want to move and the coordinate you want to move to.

Sending the following `moveToCoord` command will start moving UFOs 1, 2, and 3 towards the coordinate (75,25).

```
{
    "Command": "moveToCoord",
    "Ufos": [1, 2, 3],
    "Coord": {
        "X": 75,
        "Y": 25
    }
}
```

If you send multiple `moveToCoord` commands during one tick

buy

To buy additional UFOs you can send the `buy` command with the amount of UFOs that you want to buy and the ID of the planet where you want your new UFOs to spawn. The price of a UFO is **100000** credits.

There are some constraints:

1. The buy command will be ignored if you try to buy more UFOs that you can afford.
2. The buy command will be ignored if you try to buy UFOs on a planet that you have not conquered.

If you have no conquered planets, you can still buy UFOs using **-1** as the planet ID. This will spawn the UFOs at random coordinates in the universe.

Sending the following `buy` command will spawn 3 UFOs at planet 42.

```
{
    "Command": "buy",
    "Amount": 3,
    "Planet": 42
}
```

Macro Commands Reference

conquer

Properties

- `Command` : Type of command (conquer)
- `PlanetId` : ID of planet you want to conquer

```
{
    "Command": "string",
    "PlanetId": "int",
}
```

moveToPlanet

Properties

- `Command` : Type of command (moveToPlanet)
- `Ufos` : List of UFOs you want to move
- `PlanetId` : ID of planet you want to move to

```
{
    "Command": "string",
    "Ufos": "[int]",
    "PlanetId": "int"
}
```

moveToCoord

Properties

- `Command` : Type of command (moveToCoord)
- `Ufos` : List of UFOs you want to move
- `Coord` : (X,Y) coordinate you want to move to

```
{
    "Command": "string",
    "Ufos": "[int]",
    "Coord": {
        "X": "int",
        "Y": "int"
    }
}
```

buy

Properties

- `Command` : Type of command (buy)
- `Amount` : Quantity of UFOs that you want to buy
- `PlanetId` : ID of planet where your UFOs will spawn

```
{
    "Command": "string",
    "Amount": "int",
    "Planet": "int"
}
```

Micro

> Control your UFOs and battle against enemy UFOs.

Game Mechanics

Players control bots using 2 commands: `move` and `shoot`.

Moving

The `move` command moves the bot in the specified `direction` at the specified `speed`.
The movement is based on the polar coordinate system, `direction` being the *angle* and `speed` being the *radius*.

Parameters:

- `direction`: Value between **0** and **360**.
- `speed`: Value between **0** and **10**.

Shooting

The `shoot` command shoots a lazer from the origin of the bot in the specified `direction`.
Unlike moving, shooting has a fixed projectile speed.
The movement is based on the polar coordinate system, `direction` being the *angle* and the fixed projectile speed being the *radius*.
Shooting also has a *cooldown* of **0.5** seconds meaning that you can only shoot twice per second.
Each shot does **2** points of damage. There is no friendly fire.

Parameters:

- `direction`: Value between: **0** and **360**.

Hitpoints

A bot starts with **20** `hitpoints` and is destroyed when it's `hitpoints` reaches `0`.

Position

All bots have a `position` "**x,y**". This represents the bot's current position on the arena.
The arena has a specified `width` and `height`. The top left corner of the arena represents "**0,0**" and the bottom right corner of the arena represents "**width,height**".
Your bot can only move within the bounds of the arena. There is no collision between bots.

The Laz0r Fence

The arena is surrounded by a lazor fence. If any bot touches this fence, they will immediatly be destroyed (**0** `hitpoints`).
In order to keep the games short, the arena will start shrinking after a fixed amount of time and will keep shrinking until the battle is over.

Winning Condition

Last man standing: if all your bots are destroyed you lose.

Scripting

Game Loop

The game loop of any basic micro bot is as follows:

1. Read `JSON` formatted game state (input) from micro using `stdin`
2. Bot logic (where the magic happens)
3. Write `JSON` formatted commands (output) to micro using `stdout`

Note: The input and output are in JSON format

Python example:

```
while True:
    input = input()
    # Bot logic
    print(output)
```

Reading input from Micro

The input received from micro has the following format:

```
{
    "tick": "integer",
    "arena": {
        "height": "integer",
        "width": "integer"
    },
    "player": "string",
    "players": [
        {
            "id": "integer",
            "name": "string",
            "ufos": [
                {
                    "id": "integer",
                    "name": "string",
                    "hitpoints": "float",
                    "position": {
                        "x": "float",
                        "y": "float"
                    }
                }
            ]
        }
    ],
    "projectiles": [
        {
            "position": {
                "x": "float",
                "y": "float"
            }
            "direction": "float",
        },
    ]
}
```

Input example:

```
{
    "tick": 10
    "arena": {
        "height": 1000,
        "width": 1000
    },
    "playerId": 0,
```

```
    "playerName": "player0",
    "players": [
        {
            "id": 0,
            "name": "player0",
            "bots": [
                {
                    "id": 0,
                    "name": "bot0",
                    "hitpoints": 20,
                    "position": {
                        "x": 25,
                        "y": 25
                    }
                },
                {
                    "id": 1,
                    "name": "bot1",
                    "hitpoints": 10,
                    "position": {
                        "x": 25,
                        "y": 25
                    }
                }
            ]
        },
        {
            "id": 1,
            "name": "player1",
            "bots": [
                {
                    "id": 0,
                    "name": "bot0",
                    "hitpoints": 15,
                    "position": {
                        "x": 75,
                        "y": 75
                    }
                },
                {
                    "id": 1,
                    "name": "bot1",
                    "hitpoints": 15,
                    "position": {
                        "x": 50,
                        "y": 75
                    }
                }
            ]
        }
    ],
    "projectiles": [
        {
            "position": "23,34",
            "direction": "32",
        },
        {
            "position": {
                "x": 25,
                "y": 25
            },
            "direction": 32
        }
    ]
}
```

Writing output to Micro

The output sent to micro has the following format:

```
{
```

```
    "commands": [
        {
            "id": "integer",
            "move": {
                "direction": "float",
                "speed": "float"
            },
            "moveTo": {
                "x": "float",
                "y": "float",
                "speed": "float"
            },
            "shoot": {
                "direction": "float"
            },
            "shootAt": {
                "x": "float",
                "y": "float"
            }
        },
    ]
}
```

Output example:

```
{
    "commands": [
        {
            "id": 0,
        },
        {
            "id": 1,
            "move": {
                "direction": 0,
                "speed": 10
            }
        },
        {
            "id": 2,
            "shoot": {
                "direction": 90
            }
        },
        {
            "id": 3,
            "move": {
                "direction": 180,
                "speed": 10
            },
            "shoot": {
                "direction": 270
            }
        },
    ]
}
```