DMSC AI Tournament

Third edition

The day’s program

Schedule:

- 13:00 - 13:30: Game presentation
- 13:30 - 18:00: Work individually on your AI (use the DMSC office space)
- 18:00 - 19:30: Tournament (and food!) in Curie/Holmes

How we run this day:

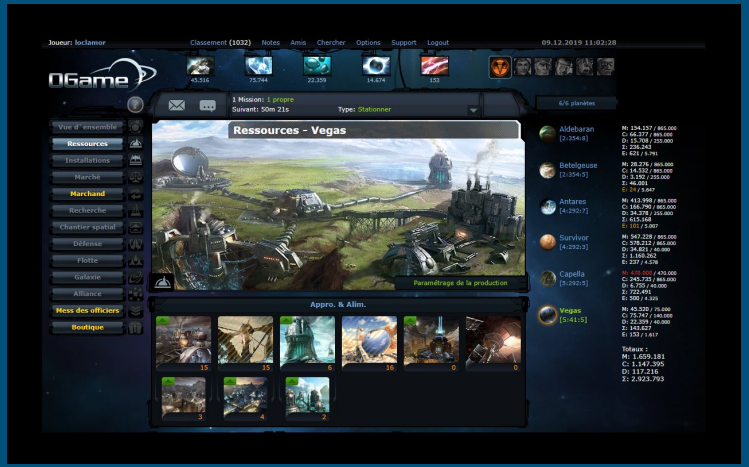
- Have fun with the challenge (don’t stress out Mads!)
- It is (most probably) possible to cheat, please show good sportsmanship

About the game

- Inspired by:

  Starcraft, OGame, Valheim, …

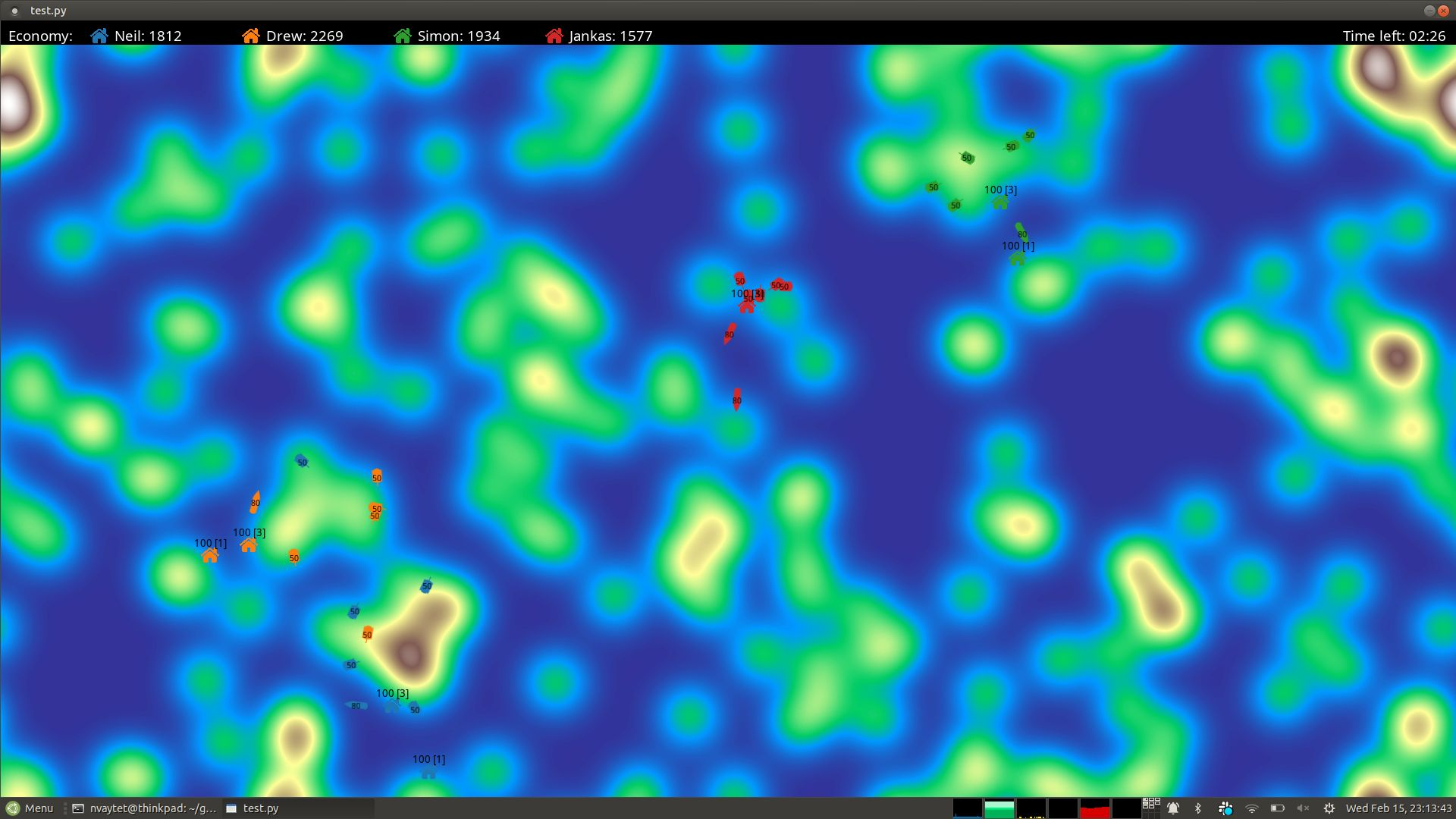
- Implementation reused almost nothing from “Quest”

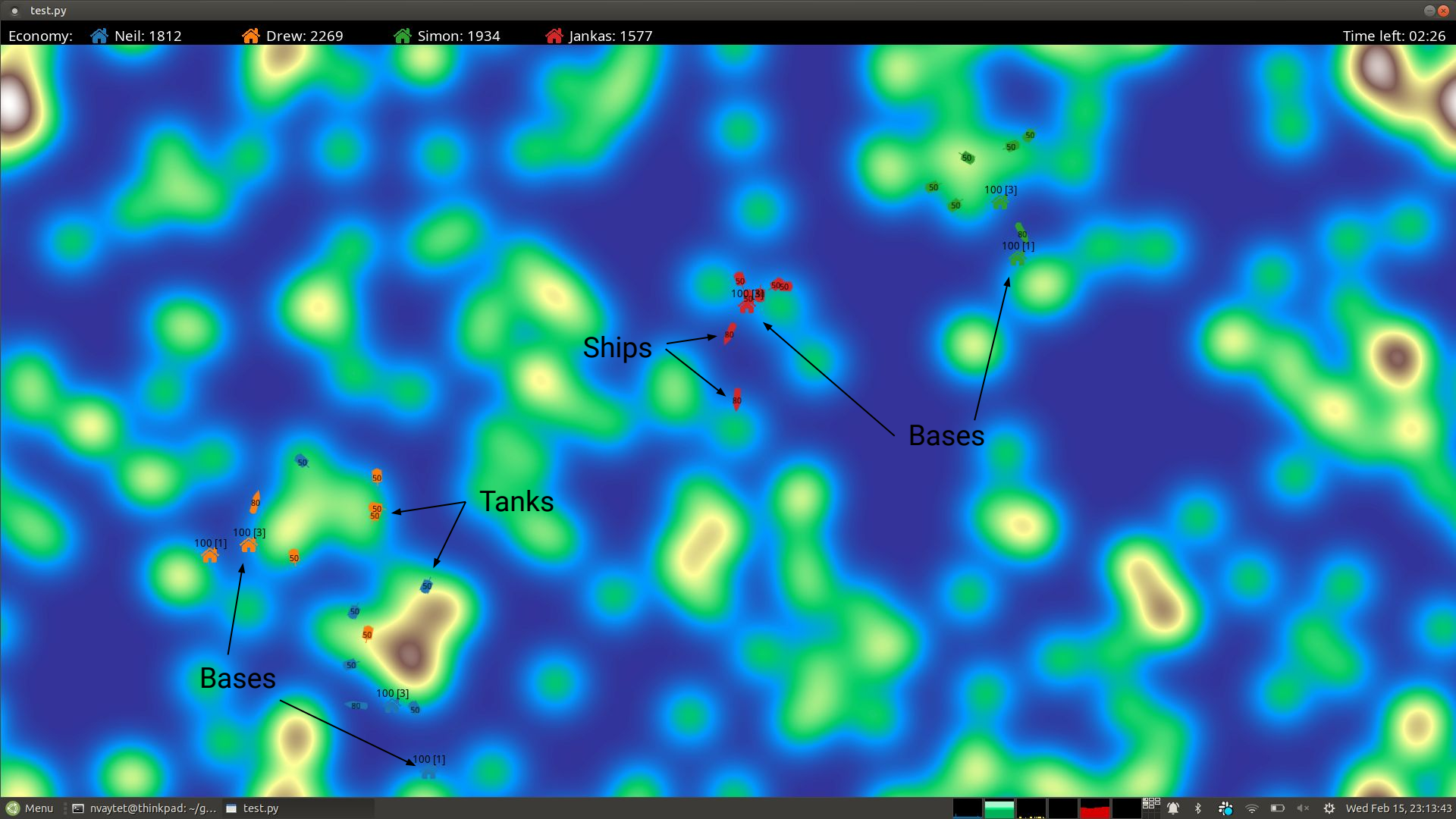


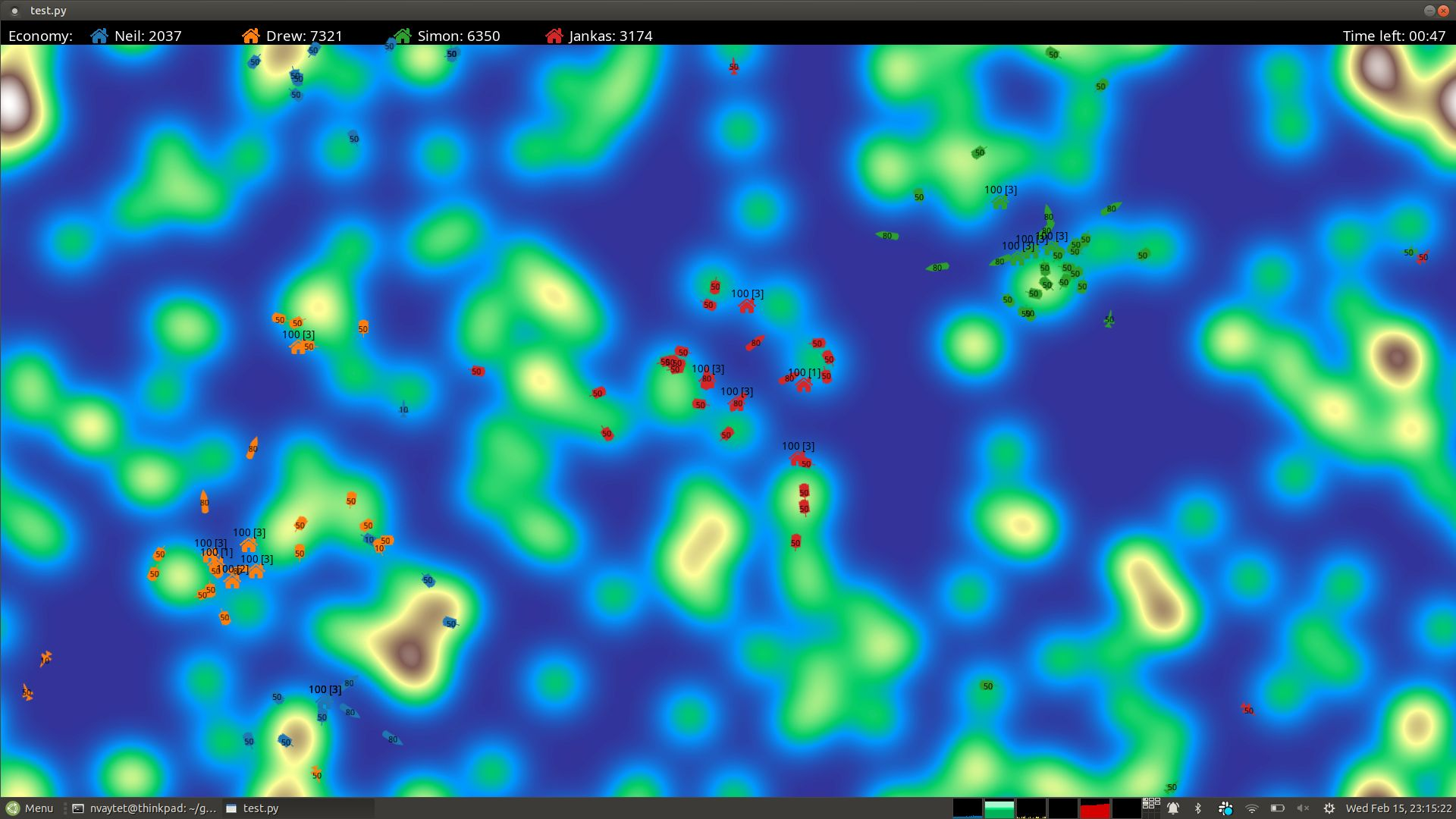
- Python with `pyglet` graphics (~1300 lines, 0 unit tests)

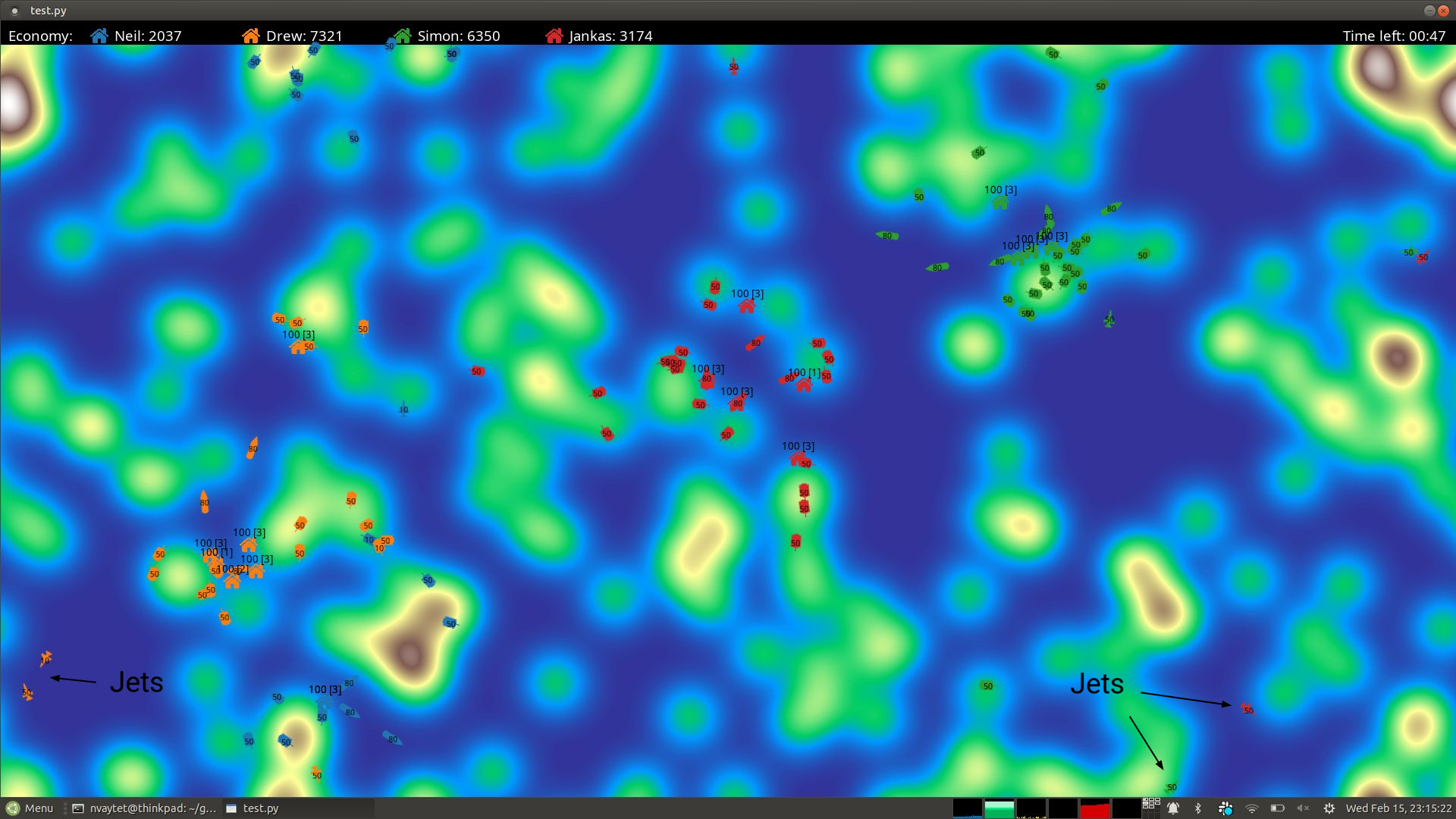
- Install via `pip install .`


LONDON
BRUSSELS
PARIS
BERLIN
VIENNA
HAMBURG
DANZIG
BRESLAU
COLOGNE
FRANKFURT
NANCY
STRASBURG
MUNICH
INNSBRUCK
SUPREMACY


test.py
Economy:
Neil: 1812
Drew: 2269
Simon: 1934
Jankas: 1577
Time left: 02:26
100 [3]
100 [1]
80
50
Menu
nvaytet@thinkpad: ~/g...
test.py
Wed Feb 15, 23:13:43


test.py
Economy:
Neil: 1812
Drew: 2269
Simon: 1934
Jankas: 1577
Time left: 02:26
Ships
Bases
Tanks
Bases
Menu
nvaytet@thinkpad: ~/g...
test.py
Wed Feb 15, 23:13:43


test.py
Economy:
Neil: 2037
Drew: 7321
Simon: 6350
Jankas: 3174
Time left: 00:47
Menu
nvaytet@thinkpad: ~/g...
test.py
Wed Feb 15, 23:15:22


test.py
Economy:
Neil: 2037
Drew: 7321
Simon: 6350
Jankas: 3174
Time left: 00:47
Jets
Jets
Menu
nvaytet@thinkpad: ~/g...
test.py
Wed Feb 15, 23:15:22

Game rules (1/3)

Goal:

- Mine resources to build and army
- Destroy enemy bases and eliminate other players
- All players play on the map at the same time
- Each round lasts 8 minutes

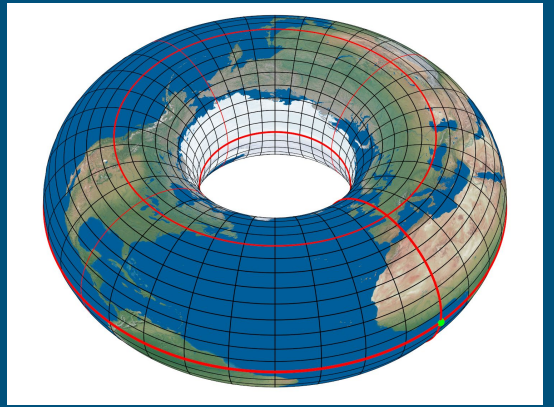
Game map:

- Dimensions: `nx = 1920; ny = 992`
- Coordinate system: lower left = `(0, 0)`, upper right = `(1920, 992)`
- Periodic boundary conditions

Game rules (1/3)

Goal:

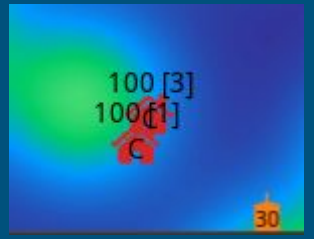
- Mine resources to build and army
- Destroy enemy bases and eliminate other players
- All players play on the map at the same time
- Each round lasts 8 minutes



Game map:

- Dimensions: `nx = 1920; ny = 992`
- Coordinate system: lower left = `(0, 0)`, upper right = `(1920, 992)`
- Periodic boundary conditions

Game rules (2/3)



Mining

- Everyone starts with 1 base, housing 1 mine
- Every timestep, each mine will extract `crystal = 2*number_of_mines`
- Crystal is used to build mines and vehicles
- Mines too close to other mines compete for resources:

```
crystal = 2 * number_of_mines / number_of_bases_inside_square_of_80px
```

Fights

- Whenever two or more vehicles or bases from opposing teams come within `5px` from each other, they will fight
- Every object hits all the others with its attack force, and it takes damage from all other objects
- No cooldown, fights are resolved (almost) instantly (in a single time step)

Vehicles

	Tank	Ship	Jet
Speed	10	5	20
Attack	20	10	30
Health	50	80	50
Cost	500	2000	4000
Can travel	On land	On sea	Anywhere
		Turns into base	

Game rules (3/3)

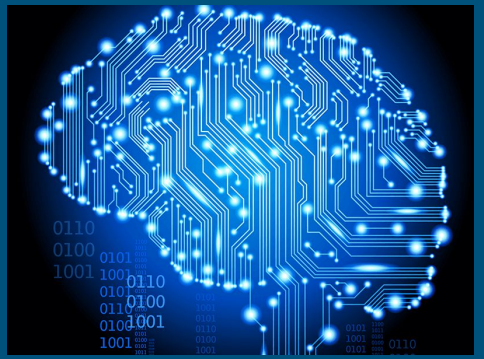
- Mine cost is x 2 for every new mine on a given base (first mine = 1000)
- Base has health = 100, mine has health = 50 (both have 0 attack)
- Vehicles move at `speed * dt` (`dt = 1/30s`)
- A ship can be turned into a new base, by calling `convert_to_base()`
- Only works if there is land in the immediate vicinity
- If a player dies, all his/her vehicles disappear

Scoring

- 1 point if you destroy a base
- If a player dies, gets a number of points equal to the number of dead players
- At the end of the round, every player still alive gets points equal to the number of dead players

Demo!

The control center - the AI (1/3)



Info (dict) it received every timestep

- One entry per player (including yourself)
- Inside each entry, a list of `'bases'`, `'tanks'`, `'ships'`, and `'jets'`

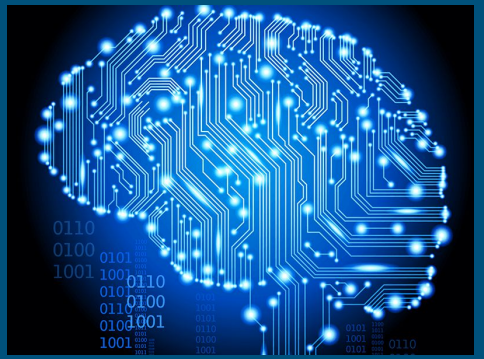
Base info

- `.x`: x position
- `.y`: y position
- `.team`: e.g. `'John'`
- `.number`: player number
- `.mines`: number of mines
- `.crystal`: amount of crystal
- `.uid`: unique id

Vehicle info

- `.x`: x position
- `.y`: y position
- `.team`: e.g. `'John'`
- `.number`: player number
- `.speed`, `.health`, `.attack`, `.stopped`
- `.heading`, `.vector`, `.position`
- `.uid`: unique id

The control center - the AI (2/3)

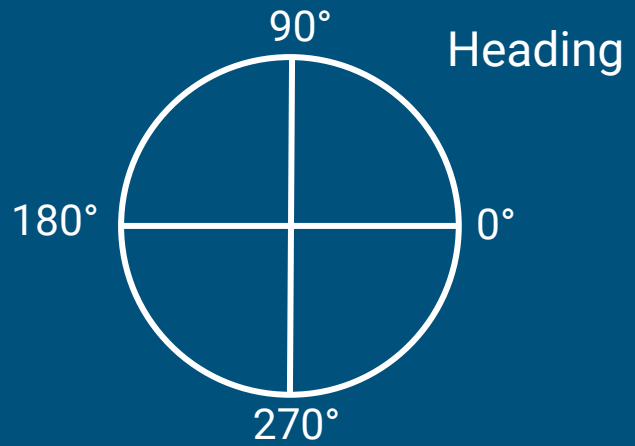


Base methods

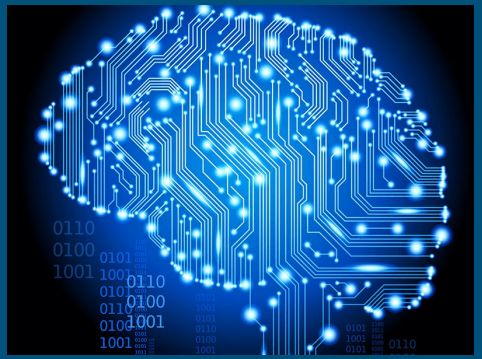
- `cost('mine')`: get the cost of an object
- `build_mine(), build_tank(), build_ship(), build_jet()`

Vehicle methods

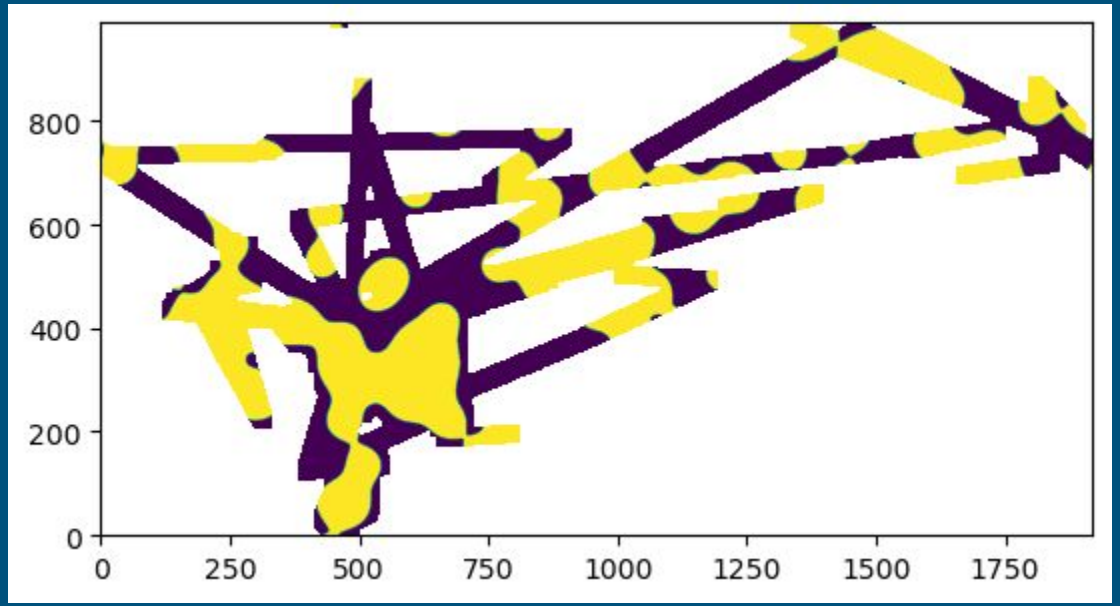
- get_position(): returns `np.array([x, y])`
- get_heading(): returns angle in degrees
- set_heading(angle_in_deg)
- get_vector(): returns `np.array([vx, vy])`
- set_vector(np.array([vx, vy]))
- goto(x, y)
- stop(), start()
- get_distance(x, y)
- convert_to_base()



The control center - the AI (3/3)



Game map is filled in for you!



- 0 = sea
- 1 = land
- -1 = no info

Template AI

Page 1

```python
import numpy as np

CREATOR = 'JohnDoe'


class PlayerAi:

    def __init__(self):
        self.team = CREATOR
        self.previous_positions = {}
        self.ntanks = {}
        self.nships = {}

    def run(self, t: float, dt: float, info: dict, game_map):

        myinfo = info[self.team]
        for base in myinfo['bases']:
            if base.uid not in self.ntanks:
                self.ntanks[base.uid] = 0
            if base.uid not in self.nships:
                self.nships[base.uid] = 0
            if base.mines < 3:
                if base.crystal > base.cost('mine'):
                    base.build_mine()
            elif base.crystal > base.cost('tank') and self.ntanks[base.uid] < 5:
                base.build_tank(heading=360 * np.random.random())
                self.ntanks[base.uid] += 1
            elif base.crystal > base.cost('ship') and self.nships[base.uid] < 3:
                base.build_ship(heading=360 * np.random.random())
                self.nships[base.uid] += 1
            elif base.crystal > base.cost('jet'):
                base.build_jet(heading=360 * np.random.random())
```

Template AI

Page 2

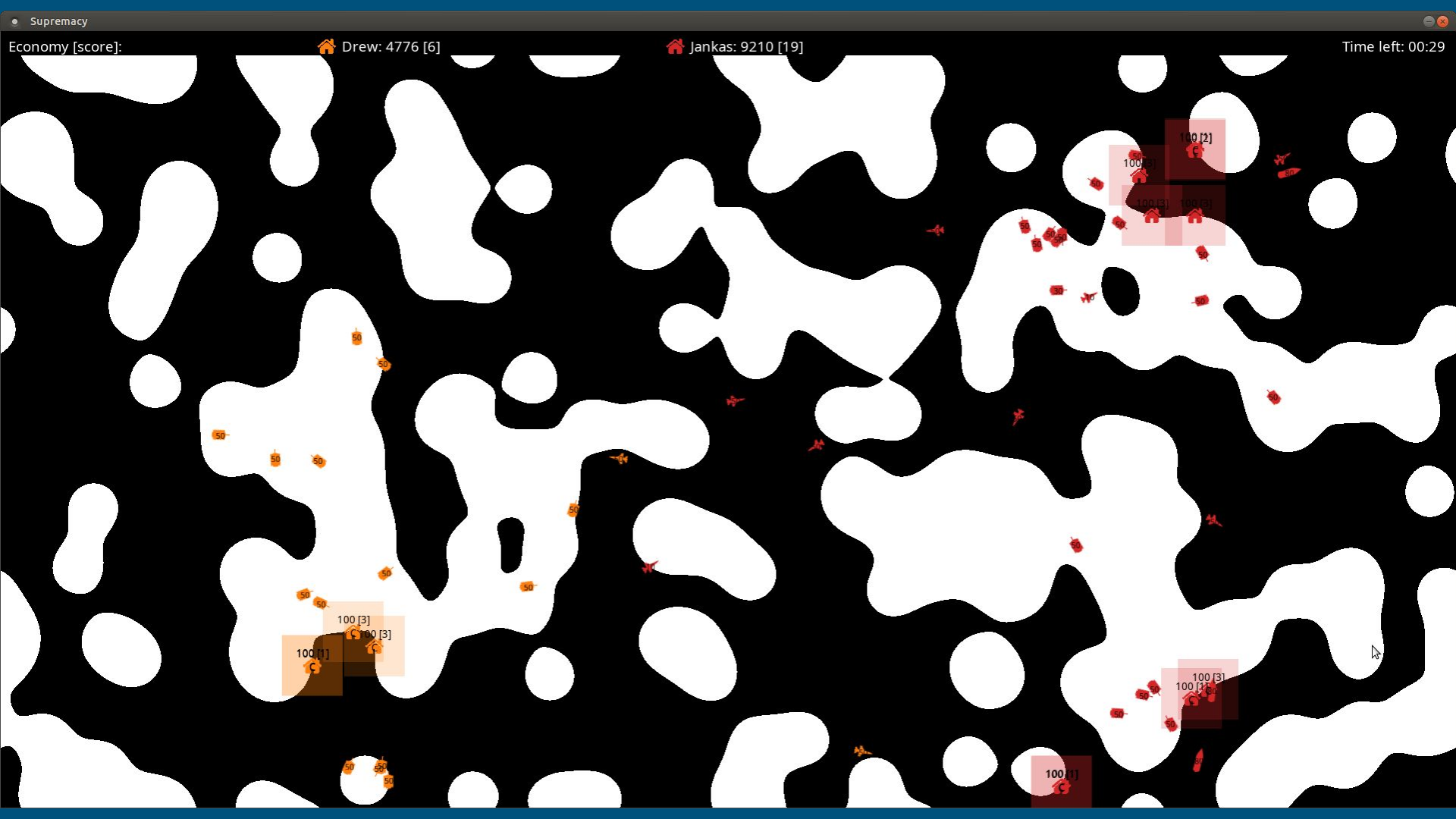
```
        target = None
        if len(info) > 1:
            for name in info:
                if name != self.team:
                    if 'bases' in info[name]:
                        t = info[name]['bases'][0]
                        target = [t.x, t.y]

        if 'tanks' in myinfo:
            for tank in myinfo['tanks']:
                if tank.uid in self.previous_positions:
                    if all(tank.position == self.previous_positions[tank.uid]):
                        tank.set_heading(np.random.random() * 360.0)
                    elif target is not None:
                        tank.goto(*target)
                self.previous_positions[tank.uid] = tank.position

        if 'ships' in myinfo:
            for ship in myinfo['ships']:
                if ship.uid in self.previous_positions:
                    if all(ship.position == self.previous_positions[ship.uid]):
                        if ship.get_distance(ship.owner.x, ship.owner.y) > 20:
                            ship.convert_to_base()
                        else:
                            ship.set_heading(np.random.random() * 360.0)
                self.previous_positions[ship.uid] = ship.position

        if 'jets' in myinfo:
            for jet in myinfo['jets']:
                if target is not None:
                    jet.goto(*target)
```

The high_contrast mode


Supremacy
Economy [score]:
Drew: 4776 [6]
Jankas: 9210 [19]
Time left: 00:29
100 [2]
100 [3]
100 [3]
100 [1]
100 [3]
100 [1]
100 [1]

Optimizing development

1. **Crystal boost:**

   Artificially increase mine yield using `crystal_boost=2`

2. **Use The ‘Pause’ Luke (experimental):**

   While the game is running, you can hit `P` on the keyboard.

   This will pause the game. You can edit your AI code.

   When the game resumes (hit `P` again), it will reload your AI module.

What's next

Get started

- Install game from `https://github.com/nvaytet/supremacy`
- Start coding!
- Having not tried any strategies further than `templateAI`, I will also participate (but will also help at the same time)

Tournament

- 10 rounds of 8 minutes (15 min hacking allowed at half-time?)
- Alliances (and betrayals) will be key!
- My hope is that every round will end up in a giant mess!

https://github.com/nvaytet/supremacy

```
conda create -n <NAME> -c conda-forge python=3.10

conda activate <NAME>

git clone git@github.com:nvaytet/supremacy.git

cd supremacy/

python -m pip install -e .

cd tests/

python test.py
```