DMSC Pybot Tournament

4th edition

Program

Schedule:

- 16:00 - 16:30: Game presentation
- 16:30 - 19:00: Work individually or in teams on your Pybot
- 19:00: Food!
- 19:00 - 20:30: Tournament!

How we run this event:

- Have fun with the challenge
- It is (most probably) possible to cheat, please show good sportsmanship

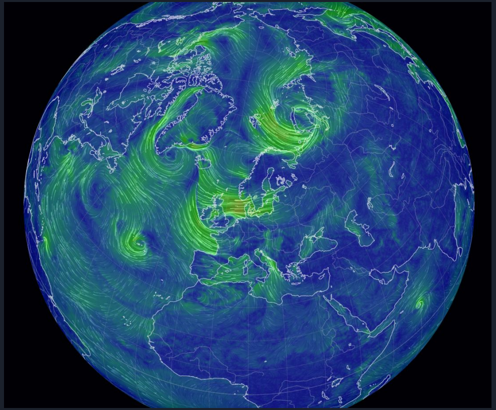
About the game

Inspired by:

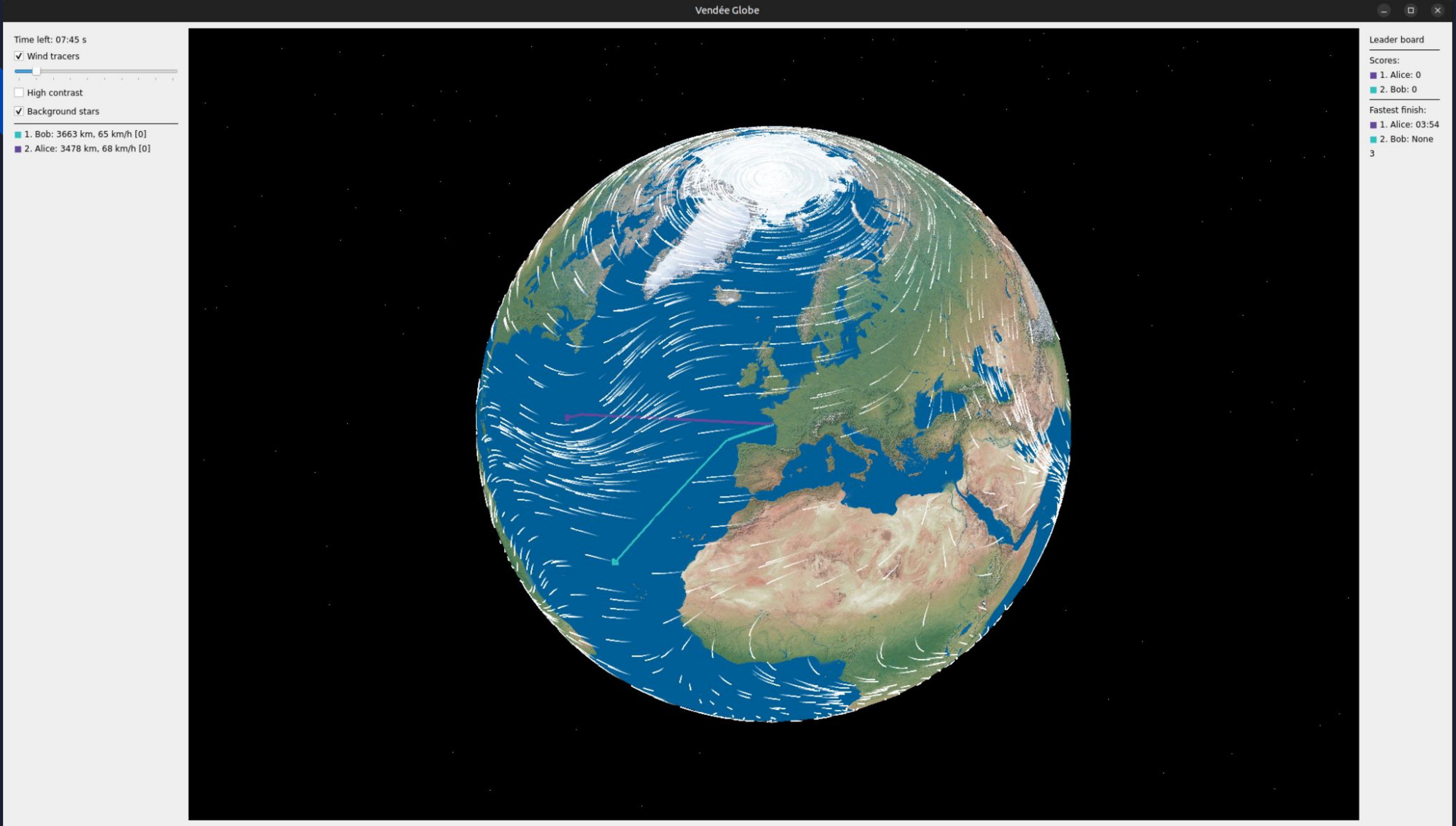
- Virtual regatta
- https://earth.nullschool.net/

Implementation:

- Python using Pygtgraph for the graphics
- ~1800 lines of code, 0 unit tests



VENDÉE
GLOBE
EN SOLITAIRE | SANS ESCALE | SANS ASSISTANCE
DÉPART LE
10 NOV 2023


Vendée Globe

Time left: 07:45 s

[✓] Wind tracers

[ ] High contrast

[✓] Background stars

🟦 1. Bob: 3663 km, 65 km/h [0]

🟪 2. Alice: 3478 km, 68 km/h [0]

Leader board

Scores:

🟪 1. Alice: 0

🟦 2. Bob: 0

Fastest finish:

🟪 1. Alice: 03:54

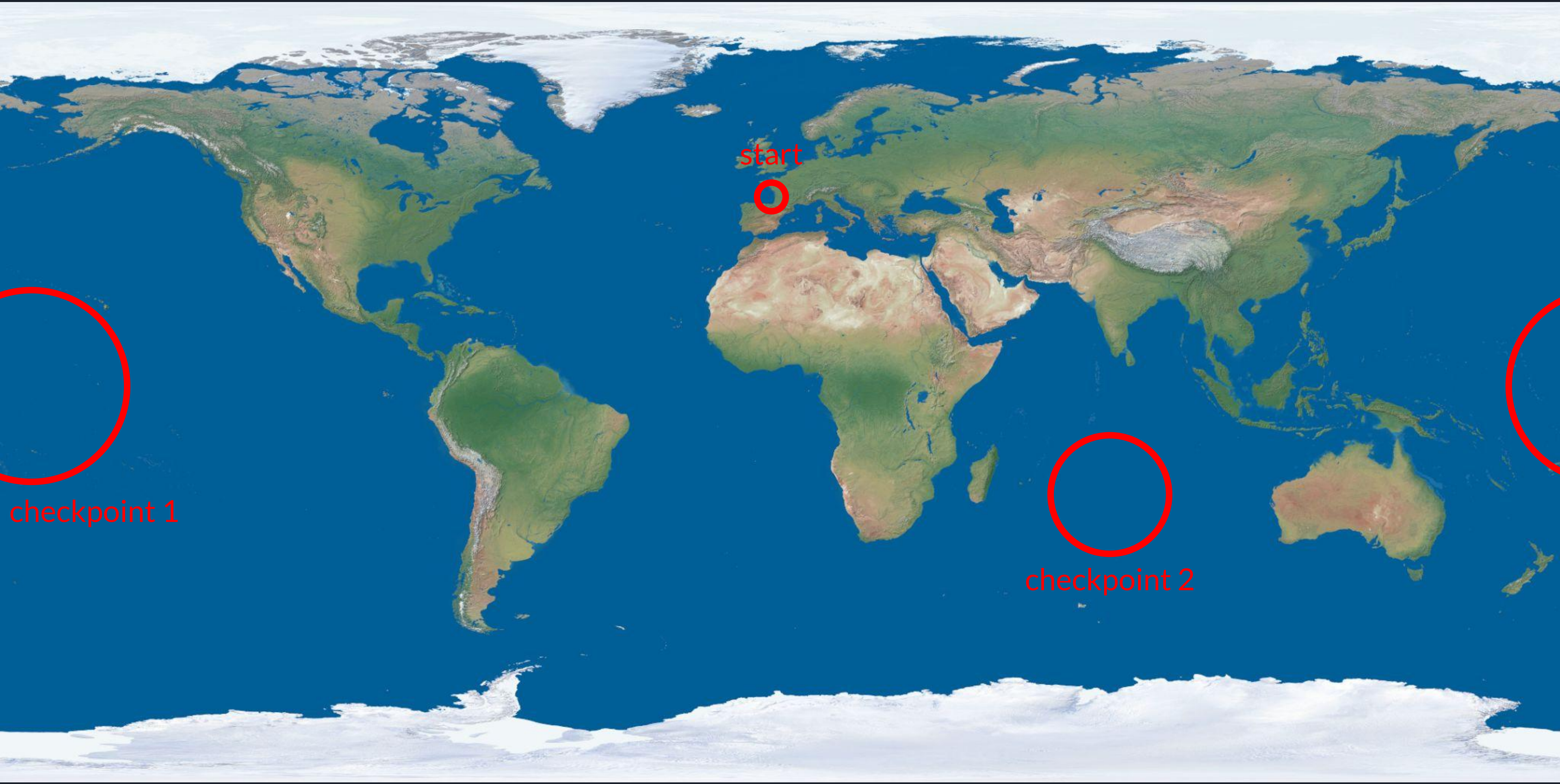
🟦 2. Bob: None

3

Game rules (1/3)

Goal:

- Sail around the world as fast as possible
- Start from les Sables-d’Olonne in France
- There are two mandatory checkpoints
- Each round has a time limit of 8 minutes (=80 days)


start
checkpoint 1
checkpoint 2

Game rules (1/3)

Goal:

- Sail around the world as fast as possible
- Start from les Sables-d’Olonne
- There are two mandatory checkpoints
- Each round has a time limit of 8 minutes (=80 days)

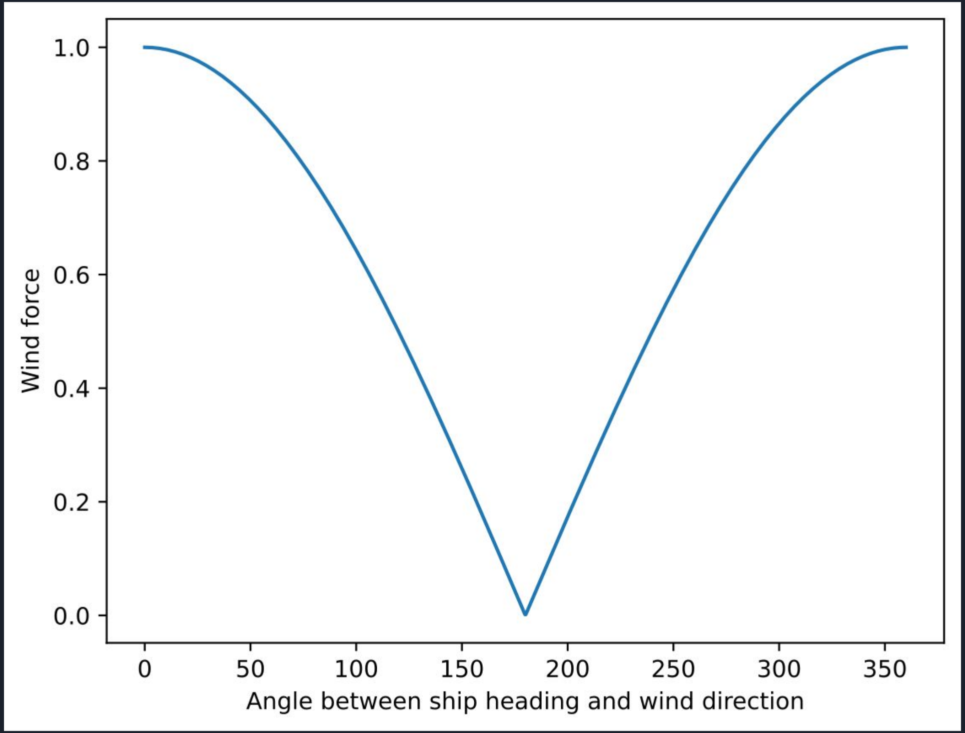
Weather:

- Only wind is a factor (no ocean current)
- Wind is generated for every game: stays static for 12 hours

Game rules (2/3)

Sailing:

- Angle between ship and wind affects ship speed
- With a wind force of 1, the ship will go as fast as the wind
- Ships will be stuck when reaching land (obviously!)
- Ships cannot crash into each other (ghosts)



Game rules (3/3)

Scoring:

- Points from 1st place to 10th place are: 25, 18, 15, 12, 10, 8, 6, 4, 2, 1 (Formula 1)
- Back home with 2 checkpoints is best (or closest to home)
- Then closest to 2nd checkpoint, then closest to 1st checkpoint

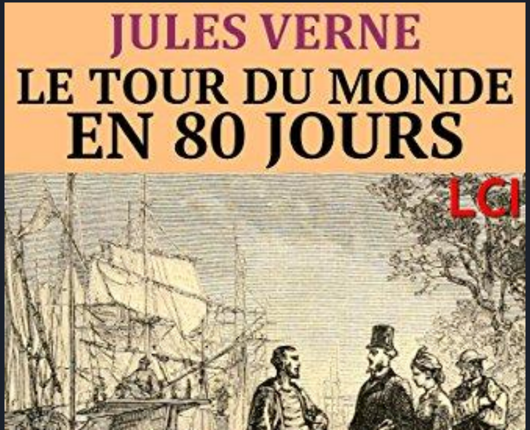
Game rules (3/3)

Scoring:

- Points from 1st place to 10th place are: 25, 18, 15, 12, 10, 8, 6, 4, 2, 1 (Formula 1)
- Back home with 2 checkpoints is best (or closest to home)
- Then closest to 2nd checkpoint, then closest to 1st checkpoint

Timing:

- 8 mins = 80 days
- 1 min = 10 days
- 6 sec = 1 day
- 1 sec = 4 hours
- 0.25 s = 1 hour



Demo!

Your **bot** (1/4)

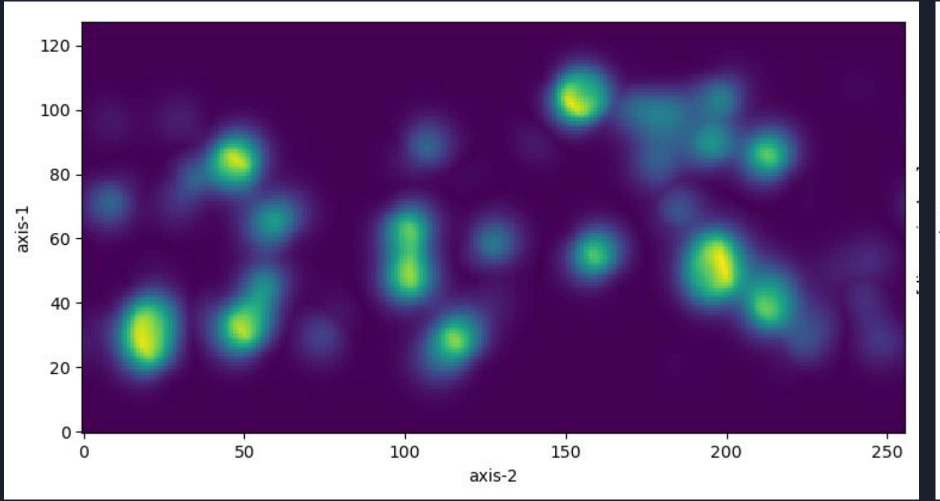
Information provided every time step:

● `t`: float:	current time in hours
● `dt`: float:	the time step in hours
● `longitude`: float:	ship longitude in degrees
● `latitude`: float:	ship latitude in degrees
● `heading`: float:	ship heading (0° = East, 90° = North)
● `speed`: float:	current ship speed in km/h
● `vector`: np.ndarray:	ship vector (u=longitude, v=latitude)
● `forecast`:	5 days of weather forecast for entire globe
● `game_map`:	world map array[lat, lon]: 1=sea, 0=land

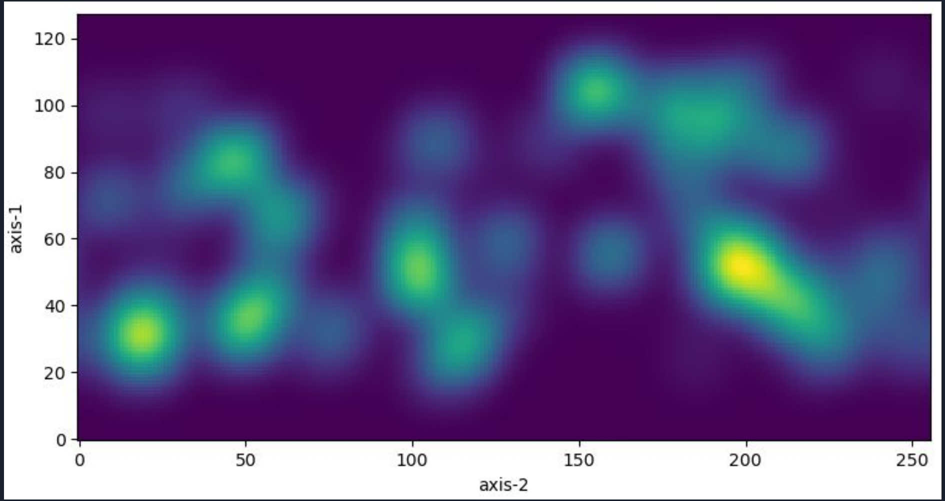
Your **bot** (2/4)

Weather forecast accuracy decreases with time:

Day 1



Day 5



Your **bot** (3/4)

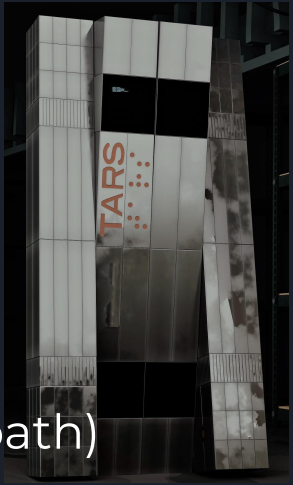
`WeatherForecast` class:

- `u, v = forecast.get_uv(lat, lon, t)` to get wind conditions for positions in space and time.

`MapProxy` class:

- `world_map.array`: np.ndarray: 1=sea, 0=land
- `lat_inds, lon_inds = world_map.get_inds(lats, lons)` to get array indices from latitudes and longitudes
- `map_values = world_map.get_data(lats, lons)` to get world map values from latitudes and longitudes

Your **bot** (4/4)



Instructions for your ship: set one of

`Location`:	a latitude/longitude to go to (shortest path)
`Heading`:	heading for the ship
`Vector`:	vector for the ship (instead of heading)
`Left`:	turn left X degrees
`Right`:	turn right X degrees

Ship speed:

- Additionally, you can control the ship's speed by choosing how much `sail` to deploy: a number between 0 and 1

Template **bot**

```python
# This is your team name
CREATOR = "TeamName"


class Bot:

def __init__(self, team: str = CREATOR):
    self.team = team # Mandatory attribute
    self.avatar = 1 # Optional attribute
    self.course = [
    Checkpoint(longitude=-45.5481686, latitude=39.0722068, radius=200),
    Checkpoint(longitude=-68.004373, latitude=18.180470, radius=10),
    Checkpoint(longitude=-80.3, latitude=9.4875, radius=10),
    Checkpoint(longitude=-79.3977636, latitude=8.6923598, radius=10),
    Checkpoint(longitude=-79.6065038, latitude=5.6673413, radius=10),
    Checkpoint(longitude=-168.943864, latitude=2.806318, radius=500),
    Checkpoint(longitude=174.900294, latitude=-16.801420, radius=10),
    config.start,
    ]
```

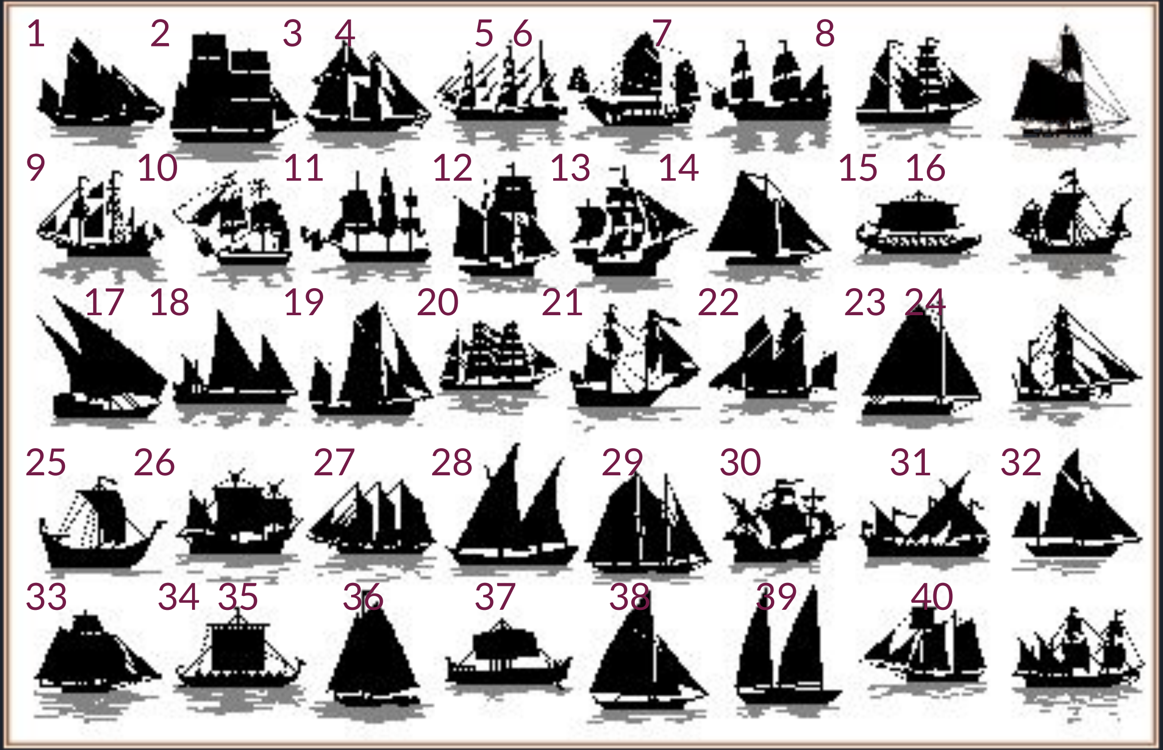
```python
def run(self, t: float, dt: float, longitude: float, latitude: float,
        heading: float, speed: float, vector: np.ndarray,
        forecast: WeatherForecast):

    instructions = Instructions()
    for ch in self.course:
        dist = distance_on_surface(
            longitude1=longitude,
            latitude1=latitude,
            longitude2=ch.longitude,
            latitude2=ch.latitude)
        jump = dt * np.linalg.norm(speed)
        if dist < 2.0 * ch.radius + jump:
            instructions.sail = min(ch.radius / jump, 1)
        else:
            instructions.sail = 1.0
        if dist < ch.radius:
            ch.reached = True
        if not ch.reached:
            instructions.location = Location(
            longitude=ch.longitude, latitude=ch.latitude)
            break
    return instructions
```

Pick your **avatar**!

Number from 1 to 40

Or

Path to a png file



Optimizing development

To speed things up, you can

1. Show a preview of your course using `course_preview`
2. Specify a starting location for your ship with `start`

   (remember to update your checkpoints!)
3. Speedup time using `speedup=2.0` (this one is a little buggy!)

Get started!

- **Register your team**: https://forms.gle/yZLGZMannpxNjGnD7
- **Clone the game** from https://github.com/europybots2024/vendeeglobe
- **Create a template repo** from https://github.com/europybots2024/template_bot
- Start coding (alone or in pairs)
- Tips:
  - Start by plotting a course (Google Maps is your friend!)
  - Wind direction is dominantly east to west

- Tournament will be 8 rounds of 8 minutes (15 min half-time tinkering)

Source: github.com/europybots2024/vendeeglobe
Template: github.com/europybots2024/template_bot

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

conda activate <NAME>

git clone https://github.com/europybots2024/vendeeglobe.git

git clone https://github.com/<USERNAME>/<MYBOTNAME>.git

cd vendeeglobe/

python -m pip install -e .

cd run/

ln -s ../../<MYBOTNAME> .

python test.py
```