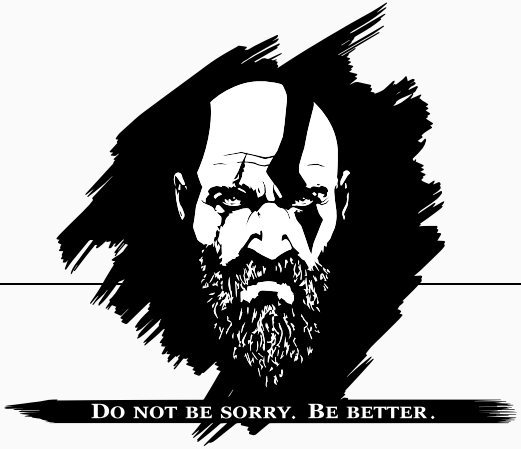
Formula One - Presentation

ACU 2019 Team



Copyright

Rules

- You must have downloaded your copy from the Assistants' Intranet <https://intra.assistants.epita.fr>.
- This document is strictly personal and must **not** be passed on to someone else.
- Non-compliance with these rules can lead to severe sanctions.

Formula One

What is it?

- **Team** project
- A **racing** game
- Implement an **AI** to move your F1 on a map
- A fun and interesting algorithmic project

Demo

Goal and problems

Your goal

- Be the first at the finish line
- Each instruction costs 1 unit of time
- Use the smallest amount of instructions

Problems

- Finding a way to the finish line
- Manoeuvering the car

About this presentation

Formulaone API

- Car
- Moves
- Map

Finding your way

- Random
- Pathfinding

Driving your car

- Granny technique
- Clone technique
- ...

Code to write

- Provided: Makefile, helper functions (`src/control.h`), viewer.
- A single function is missing: `update` (returns your next instruction), in `src/formulaone.c`.
- If you need other source files, add them to the OBJS variable in the Makefile.

Formula One API

The update function

```c
enum move update(struct car *car);
```

- Called with the **current state** of your car
- Returns your **next instruction**

enum move

- `ACCELERATE`
- `BRAKE`
- `TURN_LEFT`
- `TURN_RIGHT`
- `DO_NOTHING`

There are other combinations of these instructions.

struct car

```
struct car
{
  struct vector2 position;
  struct vector2 speed;
  struct vector2 acceleration;
  struct vector2 direction;
  float direction_angle;

  struct map *map;
};
```

`direction` and `direction_angle` is the same information represented in a different way.

Useful functions

- `car_new` allocates a new car at the map's starting position
- `car_clone` clones the given car
- `car_delete` frees the memory used by the car
- `car_move` simulates the car motion
- More in `src/control.h`

struct map

- 2D matrix of `enum floortype`
- A *map file* is a text file where:
  - `s` is *start*
  - `r` is *road*
  - `g` is *grass*
  - `b` is *block*
  - `w` is *water* (will be a block for you)
  - `f` is *finish*
- Create your own maps (*.frc* files)

Map helper function

```
enum floortype map_get_floor(struct map *map,
                             int x,
                             int y);
```

- Return the floor type
- **Do not** use `map[x][y]`: this is an interface!

Finding the way

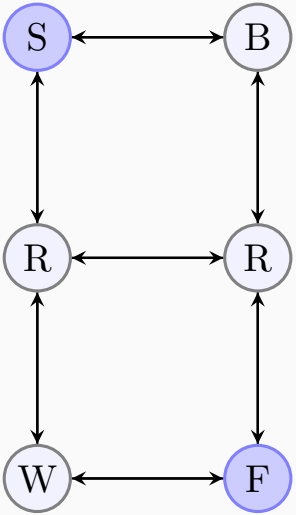
Blind

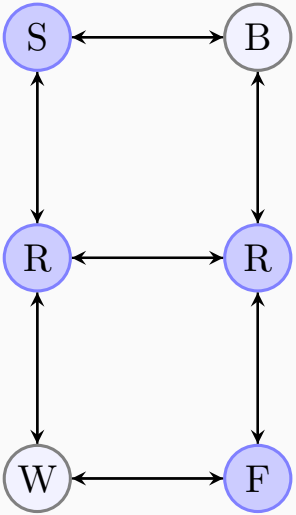
- Try moving until you find the finish line
- Notice it takes ages
- Find something else

PathFinding

- Graph representation of the map
- Locate the finish line
- Choose your pathfinding algorithm (**Dijkstra**, A*, ...)
- Keep track of checkpoints


START
ROAD
ROAD
WATER
FINISH


S
B
R
R
W
F


S
B
R
R
W
F

Driving the car

Moving: starting slowly

- Follow each elements' center
- There are many ways:
  - Granny driving

Moving: starting slowly

- Follow each elements' center
- There are many ways:
  - Granny driving
  - Clone army

Moving: starting slowly

- Follow each elements' center
- There are many ways:
  - Granny driving
  - Clone army
  - ...

Moving: next step

- Adjust your speed step by step
- Optimize the path between each element
- Optimize your pathfinding
    - Grass vs road

Moving: next step

- Adjust your speed step by step
- Optimize the path between each element
- Optimize your pathfinding
    - Grass vs road
    - Prune useless checkpoints

Moving: next step

- Adjust your speed step by step
- Optimize the path between each element
- Optimize your pathfinding
  - Grass vs road
  - Prune useless checkpoints
  - ...

Start simple

- The smartest your AI is, the harder it will be to debug
- Start **simple** and improve your AI step by step
- Finishing all the maps slowly is better than finishing half of them quickly !

Tests

How you will be tested

- Launch your AI on different maps
- Compare the number of instructions used
- Less instructions, better ranking
- Ranking with all the other students

Compare!

- Challenge your friends!
- **4 challenges** and the **Championship** organized by the ACU:
  - Tonight! (between 09:00 p.m and 01:42 a.m)

- **Bonus points** for the best ranked at each challenge
- Just **tag** your Git repository beforehand

Compare!

- Challenge your friends!
- **4 challenges** and the **Championship** organized by the ACU:
  - Tonight! (between 09:00 p.m and 01:42 a.m)
  - Tuesday night (between 09:00 p.m and 01:42 a.m)
- **Bonus points** for the best ranked at each challenge
- Just **tag** your Git repository beforehand

Compare!

- Challenge your friends!
- **4 challenges** and the **Championship** organized by the ACU:
  - Tonight! (between 09:00 p.m and 01:42 a.m)
  - Tuesday night (between 09:00 p.m and 01:42 a.m)
  - Wednesday night (between 09:00 p.m and 01:42 a.m)
- **Bonus points** for the best ranked at each challenge
- Just **tag** your Git repository beforehand

Compare!

- Challenge your friends!
- **4 challenges** and the **Championship** organized by the ACU:
  - Tonight! (between 09:00 p.m and 01:42 a.m)
  - Tuesday night (between 09:00 p.m and 01:42 a.m)
  - Wednesday night (between 09:00 p.m and 01:42 a.m)
  - Thursday night (between 09:00 p.m and 01:42 a.m)
- **Bonus points** for the best ranked at each challenge
- Just **tag** your Git repository beforehand

Compare!

- Challenge your friends!
- **4 challenges** and the **Championship** organized by the ACU:
  - Tonight! (between 09:00 p.m and 01:42 a.m)
  - Tuesday night (between 09:00 p.m and 01:42 a.m)
  - Wednesday night (between 09:00 p.m and 01:42 a.m)
  - Thursday night (between 09:00 p.m and 01:42 a.m)
  - **Championship** with the final submission on Friday
- **Bonus points** for the best ranked at each challenge
- Just **tag** your Git repository beforehand

Conclusion

Conclusion

- **Never** change the prototype of the provided functions and/or code and/or `Makefile`
- Just work in the `src/` directory
- First challenge ends in a few hours!
- Start **now!**

Newsgroup : `assistants.projets`, `[F1]` tag.

Group of : 2

Deadline : October 26, 07:42 PM

As usual:

- Your project must comply with the coding style.
- Cheating will be sanctioned.

Any questions?