

TROFF, just a tron-like.

Valentin Iovene (toogy)	Nicolas Houyengah
William Fabre (Tchii)	Sullivan Drouard
Arthur D'Avray	Karl Mire
Julien Chenglee Miyamoto	

April 22, 2013



FUCK OFF
I HATE SPECS
AND I HATE YOU FOR
ASKING ME TO WRITE OUT
THIS SHIT

Contents

1	Tasks distribution	3
1.1	So, here it is	3
2	Graphic design	3
3	GameStates	3
4	The menu	4
5	The game	4
5.1	The map	4
5.2	Players and their trails	4
5.3	Game-Over acknowledgment system	5
6	To conclude	5

1 Tasks distribution

Valentin Iovene (toogy) had the responsibility of allocating the tasks to the most appropriate members and in a fair way.

1.1 So, here it is

- Valentin Iovene: project supervisor, responsible for networking, works on almost every part of the project. Also liable for the stupidity of his team members
- William Fabre: graphic designer, responsible for the Game-Over acknowledgment system
- Arthur D'Avray: responsible for everything related to the movement of the motorcycles
- Julien Chenglee Miyamoto: responsible for the trail the motorcycles left behind them
- Nicolas Houyengah: responsible for the map. Also beer, café and pizza deliverer
- Sullivan Drouard: responsible for the game menu
- Karl Mire: responsible for the "gameStates" structure of the game

2 Graphic design

William Fabre

Let's start with one of the simplest part of this project: graphic design.

All the graphics you see in this game were made thanks to Photoshop. Adobe Photoshop is a graphics editing program developed and published by Adobe Systems. Upon loading Photoshop, a sidebar with a variety of tools with multiple image-editing functions appears to the left of the screen. These tools typically fall under the categories of drawing; painting; measuring and navigation; selection; typing; and retouching. Some tools contain a small triangle in the bottom right of the toolbox icon. These can be expanded to reveal similar tools. While newer versions of Photoshop are updated to include new tools and features, several recurring tools that exist in most versions are discussed below.

3 GameStates

Karl Mire

Every game starts off in an introduction state, then moves to a menu of some kind, a finally play begins. When you're finally defeated, the game moves to a game-over state, usually followed by a return to the menu. In most games it is

possible to be in more than one state at a time. For example, you can usually bring up the menu during game play.

The traditional way of handling multiple states is with a series of if statements, switches, and loops. The program begins in the intro state and loops until a key is pressed. Then the menu is displayed until a selection is made. Then the game begins, and loops until the game is over. Everytime through the game loop, the program must check to see if it should display the menu or simply draw the next frame. Also, the part of the program that handles events must check to see if your input should affect the menu or the game. All of this combines to make a main loop that is hard to follow, and therefore hard to debug and maintain.

4 The menu

Sullivan Drouard

This menu is composed of menu buttons. Each menu buttons has a different delegate function triggered when the user presses enter when this button is focused.

The user can browse the menu thanks to his keyword. Each button may have a subMenu or may have not.

5 The game

5.1 The map

Nicolas Houyengah

The map is a matrix of 200*150 bytes. The bytes take a different value regarding to the state of the map.

When the map is drawn, if the byte value is 0, it does nothing, if it's 1, then it colors the cell in blue (it's player1's trail), if it's 2, it colors it in yellow (it's player2's trail). If it's some other byte... well... we shall consider that we have a problem (maybe some member of the team is an asshole and don't know how to code).

5.2 Players and their trails

Julien Chenglee Miyamoto

Each player has a position on the map and a direction (North, East, South and West). This is the only information we need.

Every 10 frames, the player changes its position. When he does so, we have to change the byte value of the map matrix to the player number.

5.3 Game-Over acknowledgment system

William Fabre

Well, if the player's position is out of the born of the map matrix, he's done. If his new position (just after he moved) is on a cell of the map with a value different of 0, he's done too. Simple as that

6 To conclude

This project was great. The team was great! It probably couldn't be better. We learned a lot, this kind of exercises is really enriching, rewarding shall we say. It was a great idea, thank you.

In ACDC we can certainly trust.

