AVL CARGAME
USER MANUAL

Luis Eduardo Balaguera Ortiz
Michel Mariana García

SOFTWARE VERSION: 1.0V
28 September 2025

Content

1. Introduction

This is a small project in which we created a video game for the subject of data structures in which we demonstrated our knowledge of AVL trees. For this we learned a lot of things about pygame such as a graphic motor and use of threads in pygame to make different tasks in the same software.

a. Project objectives

Main objective in this project is learn more about software programming with different challenges on the road, for example:

- How can we use threads?
- Apply knowledge of AVL trees to create functions from scratch
- Use pygame how a graphic motor to interface
- use different elements creatively

b. Copyright statement

This is a copyright statement because in the game we use different elements (sprites) that may have copyright, so I want to say that this project is solely for educational purposes without any profit motive, and besides that I want to give a small thanks to TobyFox since some of the sprites used in the game are part of the Deltarune assets.

2. Software installation

In order to install the game, you must download or clone the GitHub repository, since we were unable to create an executable for this version.

Github link: https://github.com/ImLebo/AVL-CarNode-Game.git

You can clone the repository in VSCODE or your preferred text editor.

```
lebop@DESKTOP-O03QIM6 MINGW64 ~/Documents/Proyectos/Estructuras (main)
$ git clone https://github.com/ImLebo/AVL-CarNode-Game.git
```

Image 1.
"Example of command to clone repository"

or you can also download the project from github

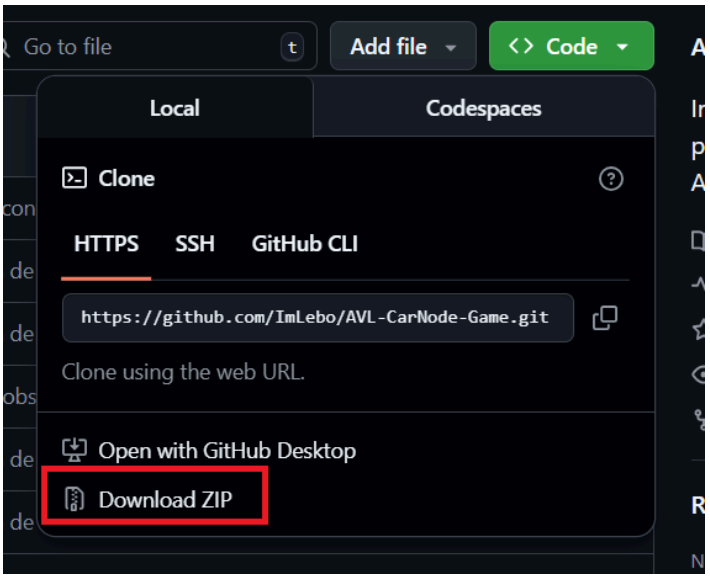

Image 2.
"Example of download the project in gitub"

If you downloaded the repository from GitHub, you have to unzip the folder to continue with the following steps.

a. Requirements of system

The only requirement for the game is that your computer can run Python 3.11 or higher. If you're not sure if your computer can handle it, you can read the Python documentation.

Link of Python documentation: https://www.python.org/doc/

b. Dependencies

The dependencies required in this project are pyplot and pygame.

To install these dependencies you can visit his official websites.

Pygame: https://www.pygame.org/news

Pyplot: https://matplotlib.org/stable/tutorials/pyplot.html

3. Run the game

If you completed all the previous steps satisfactorily, you can now start playing. To run the game you have two alternatives.

1. Run with a command

In the root folder of the game you must put this command: < python main.py >

```
PS C:\Users\lebop\Documents\Proyectos\Estructuras> python main.py
pygame 2.6.1 (SDL 2.28.4, Python 3.11.9)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

Image 3.
"Example of run the game with a command"

2. Run Python file (You require VScode)

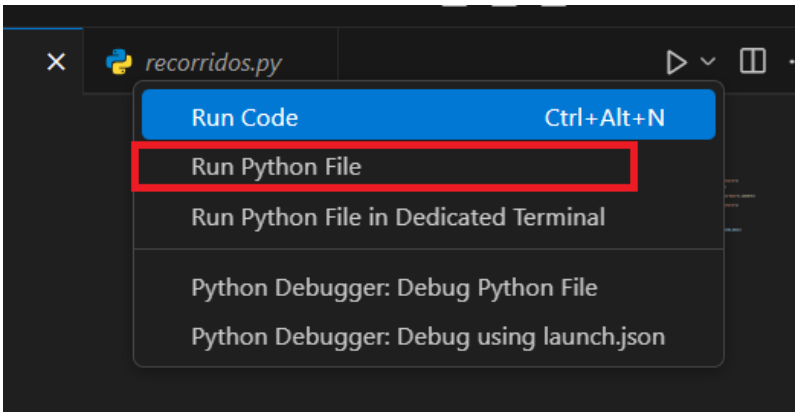

Image 4.
"Example of run the game with a Run Python File"

a. UI - User Interface

The first interface is a Menu, in this part you have a four options:

Image 5.
"Menu"

1. Jugar - Play:

In this option you can play the game, then we will emphasize on this part.

2. Opciones - Options

In this option you can modify some features of the game, in this case:

- meters (meters the car advances)
- interval in ms (milli seconds)
- distance (maximum distance from the road)
- impulse jump (More or less tall)

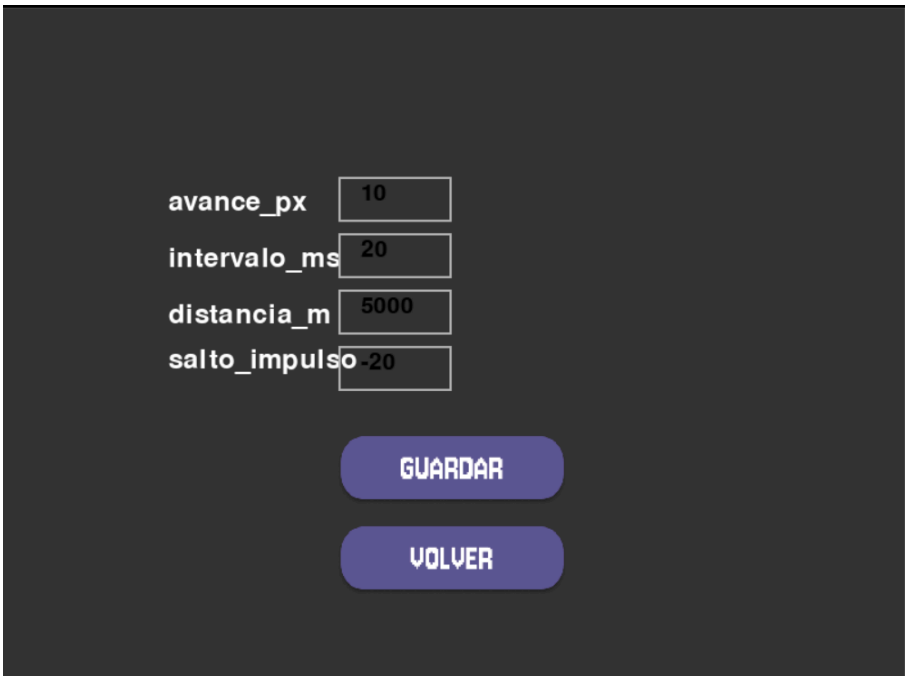

Image 5.
"Options"

3. Recorridos - Tours

In this option, you can view the AVL tree paths where the options are located on the buttons:

preorder
inorder
postorder
width

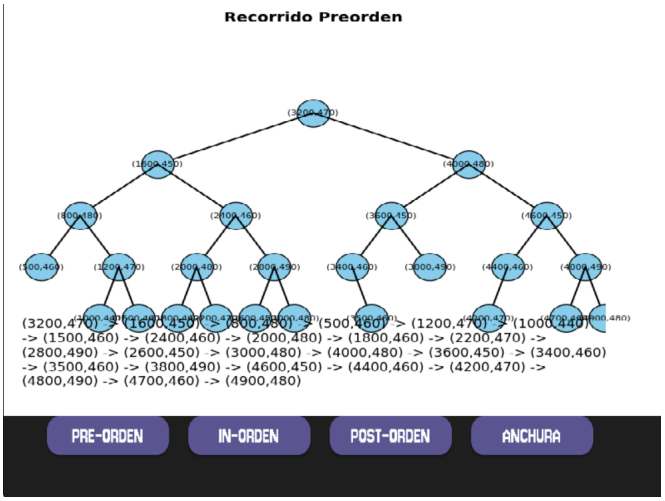

Image 6.
"Example of AVL tree paths"

4. Salir - Exit

This function, as its name indicates, is used to exit the game.

b. How to play?

When you start playing the game you have to use the up and down arrows on the keyboard to be able to move the car on the road.

Image 6.
"Buttons of the game"

You must overcome the obstacles you encounter on the road, and as they disappear from the screen, they are removed from the AVL tree. In the upper right corner, you can see the tree updating.

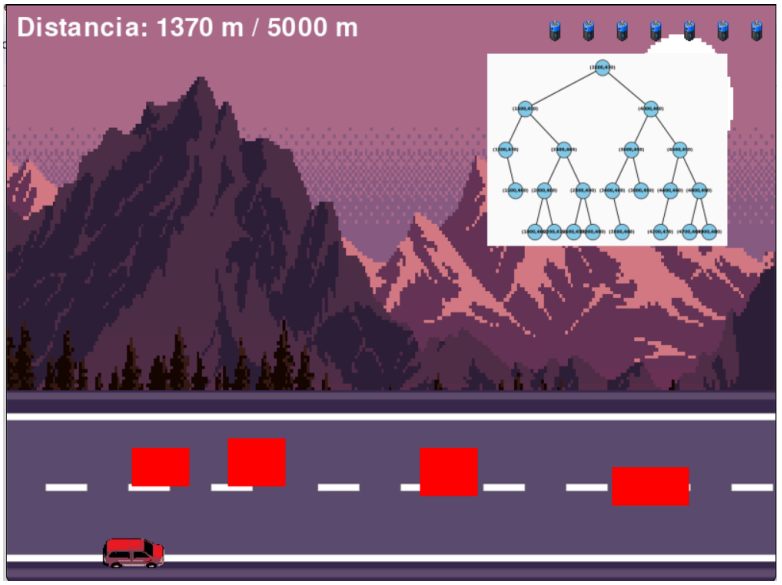

Image 7.
"Gameplay Image"

The cart initially has 10 batteries, each obstacle has a type for example 1, 2 or 3. The damage works as follows, if the car collides with an obstacle, the value of the type of obstacle it has will be subtracted, for example if it is type 3, it will subtract 3 energy.

Image 8.
“Example of damage”

Thank you very much for reading the user manual. In the repository's README, you can find a video that explains how the game works in more detail, along with a brief gameplay summary.