# Artificial Intelligence – Decision Making and Fuzzy Logic

By David Zuniga Navarro & Jose Daniel Casado Aguilar

## How to build:

1. Project is made and compiled in VS2019.
2. Open the solution file, and make sure configuration is set to x64, Debug/Release.
3. Release is recommended for a better experience.

## How to use:

* WASD - Move the camera in X and Z axis
* QE - Move the camera in the Y axis
* Mouse - While left clicking moves the camera

## What’s going on?

A terrain and a collection of vehicles is generated according to the image map in *GDP2019\_20/assets/textures/10vehicles.png*.

The vehicles are loaded with an initial velocity and have 5 sensors spread in front of them. When an obstacle (a rock in the terrain or another vehicle) gets inside one of these sensors, the vehicle steers accordingly in order to dodge it.

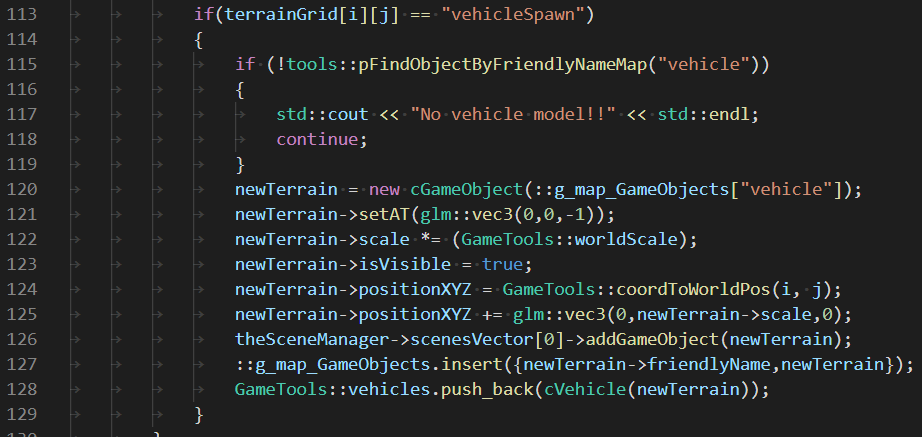
## Changing Parameters

The image can be changed for another one, keeping the name or changing the line 22 of the file *GDP2019\_20/GameTools.cpp.* The route of the file is in relation to the GDP2019\_20 project location.

## Marking Points

The *GDP2019\_20/Terrain* class deals with reading the map image and creating the objects that go in each place in the scene.

From the *Terrain.cpp:*



The *GDP2019\_20/cVehicle* class creates a vehicle with 5 sensors on it. The *GDP2019\_20/AI\_P5/Sensor* class deals with the sensor creation and logic.

In this *Sensor* class:

* The *getCollisionFuzzyValue* function calculates collisions with the sensor and returns a float from 0 to 1 depending on the distance of the collision to the vehicle.
* The *drawSensor* function draws the sensor. It receives the collision position.

The *cVehicle* class gets the fuzzy values of its sensors and decides how to steer accordingly in its *update* function. It calculates both direction and speed using this fuzzy values.