Lab 3 - Wander, Align and BlendedSteering AI Programming for Games COMP10068

Paul Keir

Date Issued: January 25, 2022
Instructor: Dr. Paul Keir

In this week's lab we will continue to use the Raylib videogame programming library to explore dynamic steering algorithms from Sections 3.3 and 3.4 of the AI for Games book by Ian Millington (eBook is in the online library).

Edit the CMakeLists.txt file in the lab-steering3 directory as before using Notepad++. You can copy from the top of last week's CMakeLists.txt file if you haven't moved anything. Ensure your antivirus software is not removing the DLLs and binary files it finds within the Vcpkg directories.

After you have used CMake to configure and generate your Visual Studio solution, open it in Visual Studio. The scenario follows from previous labs, and involves two coloured triangular ships, with one (red) as the predator; and the other (blue) as the prey.

Steering Four (steering4.cpp)

1.. Develop the `Wander` class described on the lecture slides (and in the "AI for Games" book by Ian Millington). Apply the `Wander` behaviour to the (blue) prey ship. The following `randomBinomial` function may be useful in your definition of the `Wander` behaviour.

```
float randomBinomial()
{
  std::random_device rd;
  std::mt19937 gen(rd());
  std::uniform_real_distribution<float> dis(0.0f, 1.0f);
  return dis(gen) - dis(gen);
};
```

2.. When the prey is caught, play a sound, and respawn it at a random location on the screen. To play a sound, first create a `raylib::AudioDevice` object and a `raylib::Sound` object constructed from a file in the resources directory provided. Call the `raylib::Sounds::Play` method when a prey is caught. Try using *relative* path such as "../resources/weird.wav".

3.. If either of the ships goes off the edge of the screen, ensure they reappear at the opposite side of the screen (screen wrapping). This need only be a few lines within `Ship::draw`. You may find `std::fmod` useful.

4.. Develop an `Align` class. Combine its behaviour with that of the provided `Seek` class using another class, `BlendedSteering`, to make the predator (red) align towards and seek the prey. Ensure the prey has a lower speed than the predator.