Artificial Intelligence for Video Games

End course project (A.A. 21-22)

Professors: Dario Maggiorini, Davide Gadia

Candidate: Alessio La Greca, 990973

**Index**

**Introduction**

**Chapter 1: The dungeon generation**

* 1.1: Parameters for customizing the dungeon

**Chapter 2: The graph generation**

**Chapter 3: Flocking behaviour**

**Conclusions: what could have been done better**

**Introduction**

**In this documentation, you’ll find all the logic behind this project, that could be classified as:**

***Moving flock in a procedurally generated dungeon.***

**The main idea behind this project is divided in three steps, and each of them uses one (or more) of the many topics covered in the Artificial Intelligence for Video Games course of the University of Milan:**

1. **Procedural Content Generation of a dungeon using Space Partitioning Algorithm.**
2. **From Map to Graph: how to convert the dungeon in a graph, that will be used to find a path from a starting room to an end room. The search is done using A\* algorithm.**
3. **Use of a flock and combination of different Steering Behaviours to simulate bats that move in a cave.**

**The project has been developed Unity (the editor version is 2020.3.25f1), and this documentation should be considered both as a manual for Professors Dario Maggiorini and Davide Gadia, and as a general documentation for whoever is interested in taking a look at this project.**

**Chapter 1: The dungeon generation**

* 1. **– Parameters for customizing the dungeon**

The reason I chose to start this project with Procedural Content Generation, and in particular the Space Partitioning Tree algorithm, is that I’m a fan of Roguelikes myself, and captivated by algorithms that generate random dungeons at each run, like the ones in *The Binding of Isaac*. But what I also like about algorithms is generality: that is, the possibility to tweak some input parameters to get valuable outputs. A good example of this is the qsort algorithm of the standard C library, which allows the programmer to specify a function used to compare two elements.

So, I wanted to achieve a good degree of freedom when generating a dungeon, specifying a lot of things that would have allowed me to obtain different dungeons not only because of the random seed used, but also because of the different input parameters. So, I came up with the following parameters, that can be customized in the Unity Editor.