Game of Cats Report

In this project we simulate the activity of cats on a mountain. The cats have independent features. For example, they have different age; they can be kitten or adult cats. They can be male or female (but when constructing a new object we use a default undetermined value). The cats have different personalities - they can be friendly, aggressive or lazy. The mountain environment they populate contains 2 obstacles.

We model the cats using objects and add 3D graphics to the simulation. We study how they move along the mountain and how they explore their environment using the Moore neighborhood. The Van Neumann functionality is also implemented. However, in what follows we will study the Moore neighborhood.

Figure 1 captures the beginning of the simulation, where the Cat objects are initialized and placed on a mountain terrain, whereas Figure 2 depicts the end of the simulation, more precisely the 10th iteration, where the Cats have stopped exploring their environment via the Moore neighborhood.

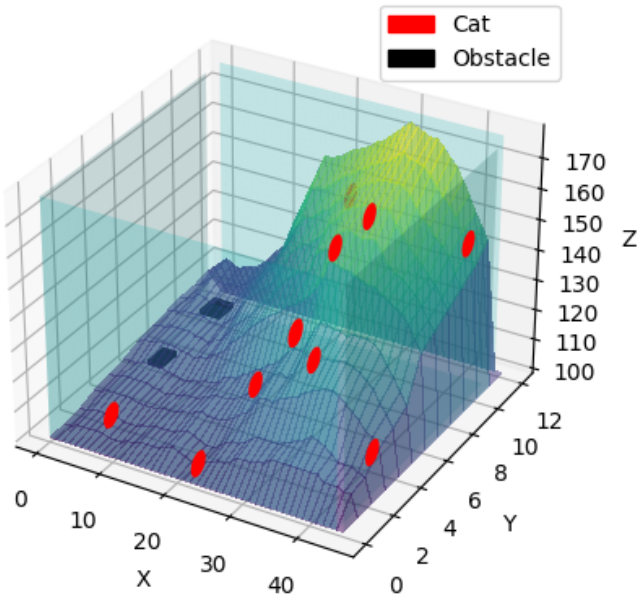

Figure 1: Beginning of the simulation. 10 Cat objects have been initialized. The environment contains 2 3D shapes that act as obstacles in the environment.

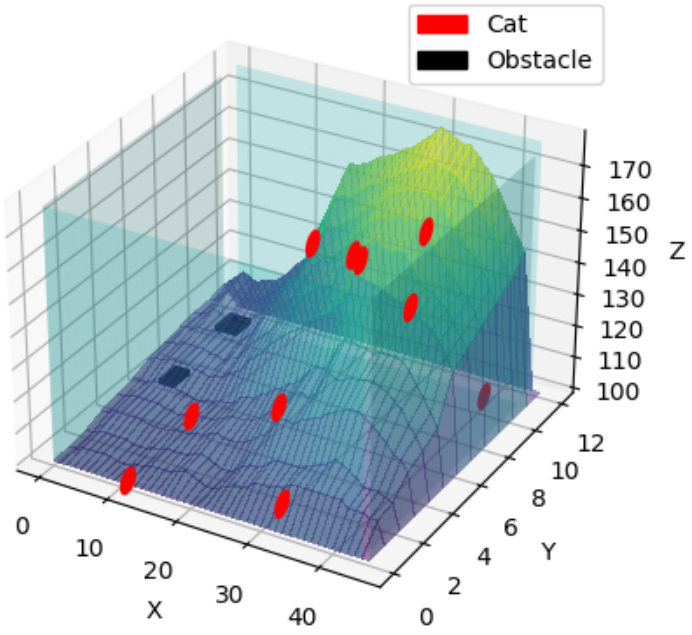

Figure 2: End of the simulation. The 10 cats have stopped exploring their environment.

The simulation was created using the following steps.

- First, we produced the **terrain** by reading a 2D matrix from a file into a pandas dataframe. The terrain depicts a mountain. Apart from the shape illustrated above, we also experimented with a more challenging shape, in the sense that it contains multiple edges. This shape is depicted in Figures 3 and 4, which show the beginning and the end of the simulation, respectively.

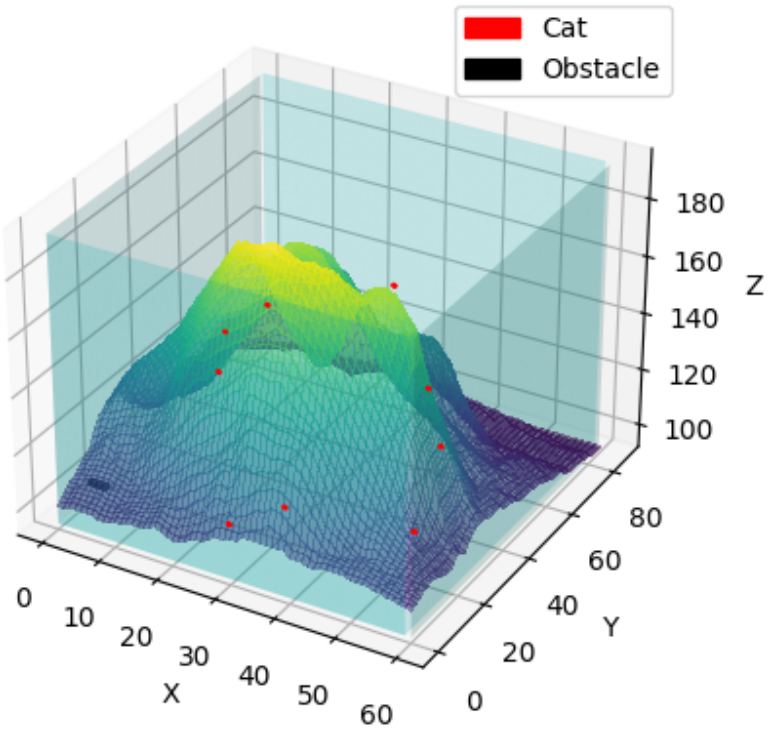

Figure 3: Beginning of the simulation. 10 Cat objects have been initialized. The environment contains 1 3D shape that acts as an obstacle in the environment.

- We then added **boundaries** to the 3D environment which we created with the help of meshes in order to prevent the cats from leaving the environment.
- We added the **cats** which were modeled as Ellipses. To generate the cats, we randomly sampled 10 cats from the points found on the surface of the mountains.
- We added **cubic obstacles** and generated them using matplotlib's Poly3DCollection.

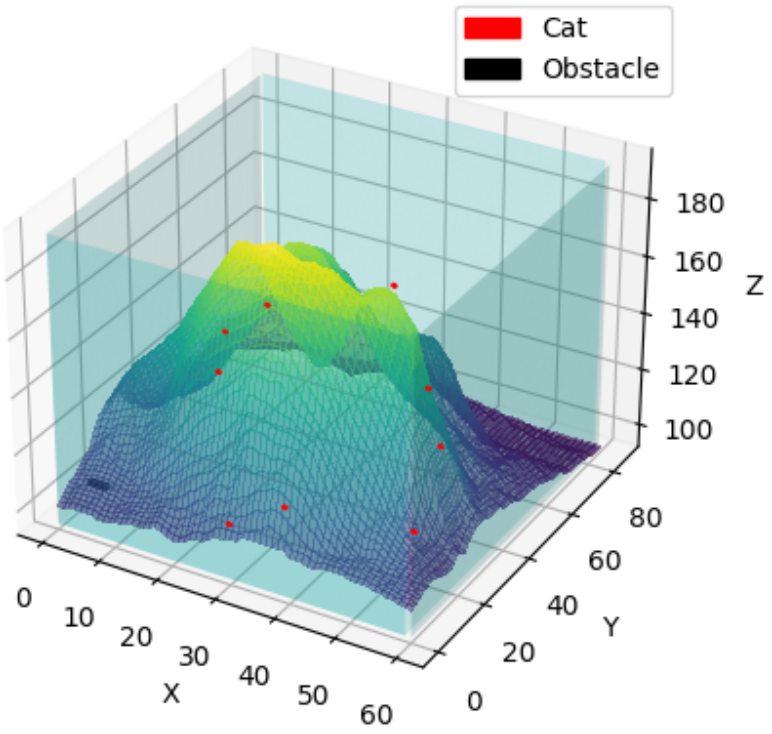

Figure 4. End of the simulation. The 10 cats have stopped exploring their environment.

- Next, we simulated the walk of the cats by exploring the Moore neighborhood as follows: we enumerated all the cells belonging to the Moore neighborhood and as soon as we found a Moore cell that is valid (is found on the grid), we drew the cat and added it to the next population of cats (the subsequent iteration). If we don't find a neighborhood that fulfills this condition, we keep the current position of the cat through the subsequent iteration.

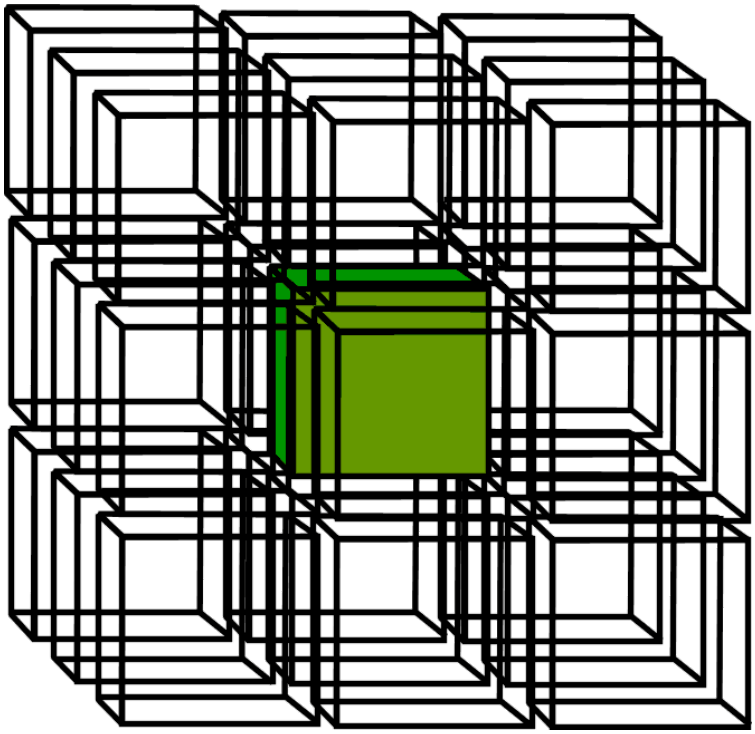

Figure 5: Moore neighborhood in 3D.

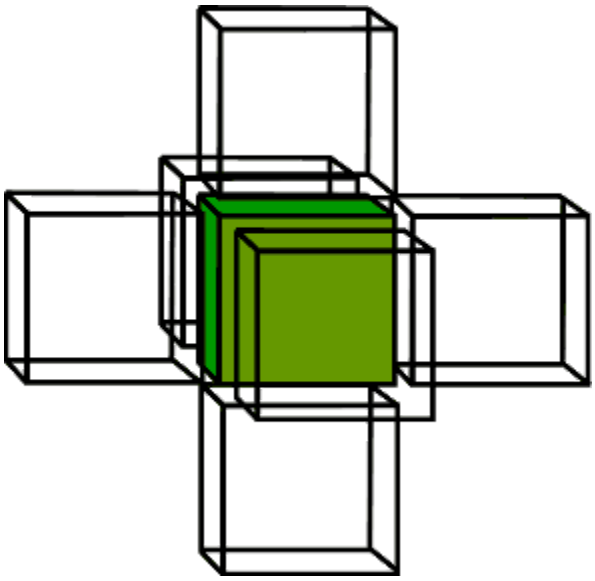

Figure 6: Van Neumann neighborhood in 3D.