



Practice Activity: Working with Distance Fields

For this week's mini-project, you will simulate zombies chasing humans. To guide the movement of the zombies and humans, you will compute a discrete approximation of both the distance to the nearest human and the nearest zombie. This discrete approximation, known as a *distance field*, is a 2D list with the same dimensions as the underlying grid. The entries of the distance field will contain an estimate of the distance for each cell in the grid. To help you gain some hands-on experience with distance fields, we will guide you through computing a very simple example of a distance field.

Distances

Many of you are probably familiar with the concept of Euclidean distance from high school geometry. Given two points (x_0, y_0) and (x_1, y_1) , the Euclidean distance between the points is $\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$. The Euclidean distance corresponds to the length of a straight line segment connecting the two points.

For this activity, we will measure distance in a different way that is often useful when working with grids. The Manhattan distance between two points (x_0, y_0) and (x_1, y_1) is $|x_0 - x_1| + |y_0 - y_1|$. (Here the expression $|z|$ is the mathematical notation for the absolute value of z .) This distance is called the Manhattan distance since it models the distance that a taxi-cab would need to drive to travel between the two intersections due to the rectangular layout of city blocks. In particular, the taxi-cab would be restricted to drive only north/south ($|y_0 - y_1|$) and east/west ($|x_0 - x_1|$).

Distance fields

The distance fields that you will compute in this week's mini-project are essentially Manhattan distances since the zombies are restricted to move only up/down and left/right. Moreover, since we are working on a grid, the resulting distances will all be non-negative integers. In the absence of obstacles, we can estimate the distance to the closest zombie by taking minimum of the Manhattan distance (computed using the formula above) to each zombie.

Your task for this activity is to complete [this template](#) by implementing three functions `manhattan_distance`, `create_distance_field`, and `print_field`. The function `manhattan_distance` returns the Manhattan distance between two cells in the grid. The function `create_distance_field` computes a distance field that contains the minimum of the Manhattan distances to the entities in `entity_list`. Finally, `print_field` prints a given distance field in a human readable form in the console.

Once you have attempted the practice activity, you are welcome to consult our solution [here](#). You might consider experimenting with the size of the grid and the position of the entities to gain a better understanding of how distance fields behave. For example, cells that contain an entity should always have a distance of zero.

For those of you wondering why we don't take this approach in the mini-project, the presence of obstacles in the grid makes computing the Manhattan distance more complicated in practice. (For example, taxi-cabs in Manhattan have to drive around Central Park.) The mini-project will use breadth-first search to compute this distance field efficiently even in the presence of obstacles.

[Mark as completed](#)