Write a basic minesweeper game. In minesweeper you search for mines in a playing field by entering x and y coordinates. Don't worry if you don't know minesweeper, we will guide you through the process step by step.

Create a class Mines. A Mines object has the following properties:

- x\_coordinate

- y\_coordinate

- hit (boolean)

The constructor only receives the x and y coordinates. The hit property is then False.

Write a function hit that sets the hit variable to True.

Write a function is\_hit that returns whether the mine has been hit or not.

Create a class Field. Imagine a Field object as a square with points that each have an x ​​and y coordinate. Since a field is square, it has the same height as width. The top left point has x coordinate 1 and y coordinate 1. The bottom right point has x coordinate N and y coordinate N, with N the height of the field.

A field has the following properties:

- height => the height is a minimum of 3 and a maximum of 5. Throw an Assertionerror if you want to use a different range to initialize the field.

- mines: a list of Mines objects; there can be at most 1 Mines object with a specific x and y coordinate value. The x and y coordinate of a Mines object in this list must be in the interval [1,height].

You provide both of these properties as parameters in the constructor. Check the conditions against assert statements.

Create a function hit\_mine that takes an x ​​and y coordinate and returns whether or not there is a mine on this coordinate. If there is a mine, adjust the hit status of the mine in the list to True.

Create a \_repr\_ function for the field. It looks like this:

A . for each coordinate of the field. Unless there is a hit mine, then you print an x ​​to the screen.

So, suppose you have a 3 by 3 field where no mine has been hit yet. Then you print:

. . .

. . .

. . .

Suppose you have a field with a hit mine at x = 1 and y = 1 and a hit mine at x = 2 and y = 3, then you print the following:

x . .

. . X

. . .

Create an app that asks the user how big the playing field should be.

Generate a random x and random y coordinate in your app that falls within the range of the minefield. Create a list with one my object at this position and pass this list to the constructor when creating the field object with given height.

Create a loop that repeats itself until the user enters 0.

Display the following output on the screen and ask the user to make a choice:

0. Quit the game.

1. Enter a coordinate.

At option 0 the game ends.

With option 1 you ask the user to enter an x ​​and a y coordinate.

You can use the hit\_mine function of the field class to check whether a mine has been hit.

Every time after an x ​​and y coordinate is searched, you print the field to the screen.