Animal kingdom information – Option 2

We chose extensions 1 and 2.

We started creating the game from the teacher’s sent finished game version. First, we re-wrote most of the given code to make it a bit more understandable. For example, we changed variable and function names and re-wrote the comments.

After that we started by adding a new object, which is a plant, into the game (Extension 1). Plant has a few different rules:

1. Changes its color, depending on the age
2. It tries to spread to a nearby cell
3. It dies if there are more plants nearby and it overcrowds
4. Lastly, it has a probability of dying about which we are going to talk below

Moreover, fishes can die from starvation if they don’t eat any plants.

As we stated above, we also added a probability for any object to die (Extension 2). Plants and fishes have the same chance to die from age 7 and on age 60 they have 100% chance of dying (it can be changed in the code). Bears have a bit higher chance to die: from 9 “years” up to 40 “years” where it would die immediately. It seems to us that the main advantage of our solution to this task is that the code is clean and simple. Though, the disadvantage is that it would be hard to implement a specific death probability at a certain age.

Every variable is written in constant format (capital letters) and can be changed in the top of the code. For example, all of the colors, breeding age, overcrowding rule, initial object food (if it can eat) and etc. The only constant you will not find at the top is the probability to die for which you have to find it in the code.

Everything, how the code works, should be self-explanatory but we also added comments to navigate and understand better.

The current game version should be balanced except for the plants every time surviving till the end. It was a really hard task to balance the game because the more objects and rules you add the more it is harder to balance the game. It is also worth mentioning that you might notice one strange thing which you might interpret as a bug. If you slow down the game, you can notice objects “jumping” to further cells than their neighbours. That is because game draws the grid after every cell has been updated. Moreover, cells start to update from the left-top corner and go till the right-bottom one. Therefore, if an object, on its first move, goes somewhere to the right, the cells will update with the game object and it will look as if the object made more than one move. We didn’t bother to fix this bug because it would require a lot of time and it wouldn’t be pretty solution. One quick fix would be to update the grid after each cell has updated. But then the game has around max 2 FPS (also depends on your computer) because each frame the game updates however many there are cells in width times the cells in height.

To test the game we checked if the rules worked for each individual object one by one.

Diagram

Description automatically generated

Figure 1. Flow chart of the program

Diagram

Description automatically generated

Figure 2. General flow chart of animal rules. For some animals flow of the rules may differ. For example, plants don't move