Turtles APCS1 Period 3
Jennifer Yu, Eugene Thomas, Allard Peng 2016-01-09

Final Project Proposal

El Cazador

Abstract

Spanish for the word "Hunter", *El Cazador* is a command line friendly game that incorporates all topics and tools taught in the first semester of AP Computer Science. This unique game is a combination of the timeless classics Battleship and Minesweeper. The player is the hunter in this game, who is faced with two primary objectives:

1. To catch harmless animals, or "preys".
2. To avoid dangerous animals, or "predators".

The game's graphics include a 9x9 grid with coordinate labeling. The hunter/player enters a coordinate that corresponds to one of the boxes inside the grid. The coordinate can either be the home of a prey or a predator, or close to one, which will be indicated through numbers in the box, similar to Minesweeper. The game is won when all the prey is caught, but you lose if you encounter enough predators to completely deplete your health.

Mechanics

Encounters with Animals:

Animals are encountered when the player enters in a coordinate to go to. Keep in mind that coordinates that have been previously entered cannot be entered again; any box in the grid that is not blank cannot be accessed again.

There are five possibilities for the status of the coordinate entered:

1. It houses a prey, which you catch.
2. It houses a predator, which deals you damage based on what it is.
3. It is near prey(s).
4. It is near predator(s).
5. It is near prey(s) and predator(s).

Situations where you directly land on a box with a prey or predator are rather straightforward. In the case of prey, the game would increment the amount of animals you've caught by one. In the case of predators, the game would determine what type of predator it was (elaborated on further later), and deplete your health appropriately.

In the cases where the coordinate you select is near prey or predator, elements of minesweeper are incorporated. The square will show the number of preys or predators in the eight other squares touching the current square. Whereas the pre-status of the square was blank, now post-status, the square has at least one number inside. In the case where there are no animals in the eight squares touching the square selected, then those squares will also "open", and provide numbers denoting the number of prey/predators in the eight other squares touching them.

If there are only preys, then the number that appears in the box will have a "+" preceding it. For example, if you went to the coordinates (3,5) and there was only one prey in the eight boxes surrounding the current box you were on, the box would display "+1". Likewise, if there were five preys, then the box would show "+5".

If there are only predators, the number that appears in the box will have a "-" preceding it. Using the example from before, if there was only one predator in the eight boxes surrounding the current box, then a "-1" would be displayed. Same applies to different quantities of predators.

In the situation where there are both prey and predators near you, two numbers would be displayed. Repeating the aforementioned example, if there were two preys and one predator near you, the box at coordinates (3, 5) would display "+2, -1".

Types of Animals

There are no distinguishable types of prey. Each prey you catch will increment your score by one, regardless of what it is. The different kinds of prey denoted are simply arbitrary categories meant to create a more enjoyable game.

Predators, however, have distinct traits. To take two types of predators as an example, a bear in the game will deal far more damage than a mosquito. The amount of damage dealt by each will be specified in the beginning to the player, and we hope to implement functionality to allow the player to access these classifications even in the middle of the game.

All the animals are also named. These names are, again, arbitrary and used to enhance the presentation of the project.

Graphics and Interaction

There will be minimal graphics in the game. There will be a grid printed out, which are the hunting grounds. With each action taken by the player, the grid will be updated and then reprinted with the added information. Text will also be printed out, and user input stream will be required.

Users will interact with the game through its command line interface. At the start screen, there will be minimal graphics, as well as instructions. There is also an option to learn about each animal, such as how much damage a predator deals, how likely an encounter with such an animal is, etc. In order to play the game, the user would need to enter coordinates into the game, which will then be processed as coordinates on the grid. With each entering of a coordinate, the new and updated grid is printed out, showing numbers representing the amount of predators or prey around the coordinate he/she typed in, or in the case that he/she lands on a predator/prey, then the interface will print out the details of the animal encounter. If the coordinate entered has been entered already, the user will be notified of this repeat coordinate, and prompted to reenter a different coordinate.

Users can access their current health (how many hit points they have left) and how many catches they currently have by typing in keywords. When a user's hit points reach 0, he/she will be exited out of the game with a conciliatory message. When a user has caught all the prey possible, he/she will have won the game, and the program will print out a message indicating that.

Solidifying and Showcasing

We incorporate all of the topics learned in the first semester of AP Computer Science in our game. The specific aspects of our games, and the topics they implement are detailed below:

- A Command Line Interface
  - Importing of java.util packages
- Different Types of Animals
  - Subclasses and Superclasses
  - Interfaces among predator and prey
  - Instances of a class and Instance Variables
  - Abstract Classes and Methods
  - this( ) operator
- The Hunting Ground
  - Accessing a 2-D Array
  - Creation of a final variable
  - While and For Loops for creation of a 2-D Array
  - Creation of an ArrayList of used coordinates
  - Sorting of an ArrayList for easier accession

Stretching: Added Functionality

We hope to be able to add increased functionality to our game. Here are some possible stretches to be made on our project:

- Ability to flag squares where a predator might be
  - Creation of commands within the game
- Multiplayer (2 or more)
  - The creation of multiple grids for different players
  - Taking turns between players
- Options for different sized grids
  - 16x16, 16x30, etc.
- Graphics for animals
  - Lions, tigers, and bears, oh my!