Lab Report

Michael Kim

010572235

2/22/2017

**Program Statement**

The goal of the programming assignment is to make a system for a spawning, blasting, killing and moving creatures using Linked list. The inputs to the program is numbers that user want to execute how many of a spawning, blasting, killing and moving. After spawning creatures, output should print list of creatures with numbers that user input. After input for number blasting creatures, output should indicate position where blasting placed, number of creatures get killed. After killing random creatures, output should come out remaining creatures. After moving creatures, input should output changed of position after adding with velocity that creature has. Error handling was not required at this assignment.

**Design**

In this Design section, I will explain big pictures of conducting hw2. The details of each methods and functions will be explained in Implementation.

The big picture for this hw2 was deciding order of which file I should work.

The first design decision I made is to write CreatureNode.h. After adding private values and all methods, I wrote CreatureNode.cpp. filling all methods with CreatureNode, I moved to CreatureList.h. after adding private value and parameters for given functions, I moved to CreatureList.cpp to fill up all methods. Lastly, main.cpp was filled it with proper functions.

**Implementation**

For CreatureNode.h file, I started with writing private values for creature’s information which were ID,Xposition,Yposition,Xvelocity,Yvelocity,Health,and Strength. For last, CreatureNode pointer Next was added. All the set methods were added using void type. Each parameter contains identical type of each information. Get methods were added after it, each get methods were declared with the type that needs to return to. Int was used for ID, Health, and Strength. float was used for Xposition, Yposition, XVelosity, and Yvelocity. For last, print functions and setNext was declared with void type. GetNext was declared with CreatureNode pointer type, so it can return to the pointer.

For CreatureNode.cpp file, I started with the filling constructor. All components were declared which were declared in private values. For last, CreatureNode pointer Next was set as NULL. The Deconstructor remained empty. Each set methods were component value equal to parameter. Each get methods were return to component value of corresponding component. For print methods, each component with cout statement. Setw was used for neatness.

For CreatureList.h file, int count was declared for counting number of creatures. Float Xmin,Ymin,Xmax,Ymax for maximum and minimum values for creatures position.

CreatureNode \* head pointer was declared for head pointer. After adding private values, parameters for spawning, killing, and blasting were added. No parameter added to Move and print method.

For CreatureList.cpp, constructor was just declared with head pointer that equal to NULL. Spawning method was designed with inserting head. First, declaring temp CreatureNode pointer was placed. After making new CreatureNode to temp, set methods were used for each component’s method, with parameters from spawn\_creature. setting next for temp to head, so after new node added, it can be connected to head. Head equaling to temp was last part. ID - - was added too, this will be explained in main.cpp. for killing method was placed. Two pointers were set, temp for deletion object, and prev for connecting previous node to next node from deleted node. From parameter, int id was added for selecting deletion, so if prefer ID is found from the list, it delete the Node that ID contained. If the found ID node is the head, head was equaled to Next to head.if the found ID node is in the middle, previous node next was set to temp next, so it can be connected previous to next node that deleted. Blasting method was set next, with parameters with xposition and yposition which are the position for bomb is set. There were four range from bomb place, dead, fatal, injured and hurt. If the creature is found in the dead range which is +-10, that creature’s health subtracts 100. If its in fatal range which is +-15, health is subtracted 50. Injured range +-20 was 20, hurt range +-30 was 10. After setting range for health subtracting, kill creature method was used for killing creatures with health 0. After finding ID with health 0, pass that ID to killing method parameter to kill creatures. For next, Moving creature method was set. In moving method, Adding current position to current velocity resulting with new position was set. Max and min position was set to +100 and -100. If creature goes above min and max, it stays at min and max. for last, printing creaturelist was set. After declaring pointer with head, inside the while loop that runs until pointer is NULL, use print functions from CreatureNode, and set new pointer to next node so it can go on.

For Main.cpp files, after declaring values for creatures components, spawning method was set. For xposition,yposition, random function was used with range -100 to +100. For xvelocity,yvelocity, random function was used with range -10 to +10. Strength also used with random function with 0 to 50. Health was set 100 for all creatures. For ID, initial ID was set with value of number of creatures that user want to make. As it spawn one by one, ID is decreasing. Blasting,killing and moving method was used. After finishing each method, printing list method was set.

**Testing**

First I started test with small parts. If it is operated correctly, program was built afterwards.

This program was run over a lot to catch any debugs.

It worked as I expected. Output came out corresponding of user’s input.

**Conclusion**

The overall result of the assignment came out with correct output with few statements. Project worked successful. I would try to make map for spawning,blasting,killing and moving for next time.