Enhancement Two: Algorithms and Data Structures

Charetta Frierson

Neil Kalinowski

CS 499

29 September 2019

Southern New Hampshire University

Artifact Selection

For the second enhancement I have once more chosen the zoo monitoring system from IT 145 Foundations in Application Development. For this enhancement I will be improving the overall efficiency of the system. The system helps zookeepers keep track of animal health, activities, and habitats. It uses dialog boxes to alert when animals or habitat is out of the normal range, displays information based on file, and distinguishes choices based on selection and category. The artifact was originally created in 2017 and updated September of 2019.

Inclusion and Justification

The artifact was selected to showcase skills currently asked for within the industry as a Software Engineer. The components I would like to reflect on for this enhancement will be basic data structure types and algorithms that display predefined tasks. The code will show my knowledge of inserted data structures and algorithms that fetch the given task when prompted. The text file part of the code was added to search for stated animal or habitat and broken down into two switch statements to further make the code refined.

Outcome and Objectives

As far as outcomes and objectives I feel I did complete the main components of what needed to be done. There are no major changes done as far as the outcome coverage plan at this time. I am looking to be on track with everything as described.

Reflection

There were not many challenges this week since I already began making configurations prior as both enhancement one and two coincide with one another. I would say it was much less

complicated versus the week prior. The text statements were added so the files could be read and the code was adjusted to further streamline the final outcome.

Initial Artifact

```
//ZooMonitor
import <string>;
import <vector>;
import <iostream>;
import <stdexcept>;

namespace ZooMonitor
	export class ZooMonitor
	{
	  static void main(std::vector<std::wstring> &args)
	  {
		int opt;
		Scanner *sc = new Scanner(System::in);
		ZooMonitor *zm = new ZooMonitor();
		while (true)
		{
	 std::wcout << L"Choose the option from the MENU" << std::endl;
	  std::wcout << L"------ Menu ------" << std::endl;
	  std::wcout << L"1. Monitor an animal" << std::endl;
```

```
        std::wcout << L"2. Monitor a habitat" << std::endl;
        std::wcout << L"3. Exit" << std::endl;
        opt = sc->nextInt();
        switch (opt)
        {
        case 1:
        zm->monitorAnimal();
        break;
        case 2:
        zm->monitorHabitat();
        break;
        default:
        std::wcout << L"Invalid Input" << std::endl;
        sc->close();
        break;
        }
            }
            delete zm;
            delete sc;
        }
        // habitat function.
    private:
        void monitorHabitat()
```

```
{
    int opt;
int num = 0;
std::wcout << L"------Habitat------" << std::endl;
File *file = new File(L"habitat.txt");
Scanner *sc = new Scanner(file);
while (sc->hasNext())
{
    std::wstring str = sc->nextLine();
    std::wcout << str << std::endl;
    sc++;
}
sc->close();
    delete sc;
}
void monitorAnimal()
{
    int opt;
int num = 0;
std::wcout << L"------Animals------" << std::endl;
File *file = new File(L"animals.txt");

Scanner *sc = new Scanner(file);
```

```
        while (sc->hasNext())
        {
            std::wstring str = sc->nextLine();
            std::wcout << str << std::endl;
            sc++;
        }
        sc->close();
            delete sc;          }
    }
}
```

Second Enhancement

```
#include <iostream>
#include <string>
#include <fstream>

using namespace std;
void monitorHabitat()
{
        string habitat;
        fstream file;
        file.open("habitats.txt");

        while (!file.eof())
        {
                file >> habitat;
                cout << habitat << endl;
        }
        file.close();
}
void monitorAnimal()
{
        string animal;
        fstream file;
```

```
        file.open("animals.txt");

        while (!file.eof())
        {
                file >> animal;
                cout << animal << endl;
}
        file.close();
}
int main()
{
        int x;
          cout << "Choose the option from the MENU" << endl;
        cout << "------ Menu ------" << endl;
        cout << "1. Monitor habitat" << endl;
        cout << "2. Monitor animal " << endl;
        cout << "3. Exit" << endl;
        cin >> x;
}
```

End Code

Habitats.txt

Details on penguin habitat
Details on bird house
Details on aquarium

Habitat - Penguin
Temperature: Freezing
*****Food source: Fish in water running low
Cleanliness: Passed

Habitat - Bird
Temperature: Moderate
Food source: Natural from environment
Cleanliness: Passed

Habitat - Aquarium
Temperature: Varies with output temperature
Food source: Added daily
*****Cleanliness: Needs cleaning from algae

Animals.txt

Details on lions
Details on tigers
Details on bears
Details on giraffes

Animal - Lion
Name: Leo
Age: 5
*****Health concerns: Cut on left front paw
Feeding schedule: Twice daily

Animal - Tiger
Name: Maj
Age: 15
Health concerns: None
Feeding schedule: 3x daily

Animal - Bear
Name: Baloo
Age: 1
Health concerns: None
*****Feeding schedule: None on record

Animal - Giraffe
Name: Spots
Age: 12
Health concerns: None
Feeding schedule: Grazing