School of Computing & Information Technology

CSCI251/CSCI851 Advanced Programming Spring 2019

Assignment 2 (Worth 10%)

Due 11:55pm Sunday 13^{th} October 2019.
Early diagram submission: 7pm Friday 20^{th} September 2019.

Overview

This assignment is to be implemented using object oriented programming. It involves implementing a simulation of an adventure in time, tracing the history of some artifacts.

In addition to providing code you need to:

- Submit a draft UML like class diagram indicating how your classes are structured and related.
- Submit a final report, which contains the final UML like class diagram reflecting the structure in your code, and text/tables/diagrams addressing the points below:
    1. The characteristics of the chrononauts and the impact these have. The description of the impact should be qualitative (descriptive), rather than quantitiative (numerical).
    2. Describe the 3 different kinds of chrono–pet, and the impact each has. The description of the impact should be qualitative (descriptive), rather than quantitiative (numerical).
    3. Describe the 5 different sizes of population centre, primarily the relevant population ranges, and the additional feature added for each. The relevance of those features should be noted.
    4. Describe the impact of the 5 different technological eras.
    5. Briefly describe how each of the event mechanisms work. This may be tied to the population and/or technology eras.

General notes

These are some general rules about what you should and shouldn't do.

1. Your assignment should be sensibly organised with the same kind of expectations in this area as assignment one, although you may reasonably have more files for this assignment. No memory leaks etc. too.

2. **Other than the initial command line input, the program should run without user input. In particular this means there shouldn't be pauses waiting for the user to press a key.**

3. **There shouldn't be pauses generally, please don't use sleep!**

4. Your code must compile on Banshee with the compilation instructions you provide in `Readme.txt`. If it doesn't you will likely receive zero for the coding part of the assignment.

5. You have to use classes, and should make use of encapsulation.

6. You must use inheritance. If you don't use inheritance you will lose a fair few marks.

7. You can use polymorphism, if it's appropriate.

8. You need to overload `operator++` as explained later. Other overloading, if you use it, should be sensible and appropriate.

9. You need copy constructors associated with the evolution and devolution of the location.

10. You can use your own data files for names or similar information.

1 An adventure in time

The year is 2525, and you have been appointed to supervise a team of chrononauts adventuring through time to collect information on the provenance and usage of various artefacts. The general overview is described below, but most of the details are up to you.

Your code should compile on Banshee according to instructions you provide in a `Readme.txt` file.

Once your program is compiled into the executable `AAT`, it must run as follows:

```
./AAT n m
```

where n is the number of artefacts to be traced, and m is the number of jumps to be made for each artefact. The value of n should be in the range 1 to 5 inclusive. The value of m should be in the range 1 to 10 inclusive.

Your adventure takes place across the lifetime of a population centre, which in 2525 is a metropolis but at some distant historical point was a cluster of huts in a forest clearing near a source of clean water.

You have an initial clue as to a period in the past when information on the artefact will be available. Each jump will take you to slightly before that date. You need to decide how to appropriately reduce the population centre in accordance with the size and technological era.

After each jump you step the population centre foward in time, year by year, using the overloaded ++ operator, until you find the information and an additional clue. It should typically take 5-10 years to find both of those. If no more jumps are allowed only the information needs to be gathered.

You should be reporting on all of the details of the population centre when you make a jump, so before and after. In the year by year stepping forward you can report on the technological and population levels each year, and otherwise just on events and the impact they have.

Report on the state of the chrononauts at the start and end of each jump period.

2 Simulation components

The simulation is from the perspective of the chrononauts. You need to give consistent and appropriate definitions for a class hierarchy associated with the various components of the adventure.

2.1 Artefacts, clues, and information

You should randomly generate an artefact. It should have a name and some sort of description. It should also have an estimated source time which should be relevant in determining where you jump. Across the search for an artefact it's expected you will probably need to travel through at least two different technological eras.

The clues are effectively just a single number telling how much further to jump back for the next clue.

The information is a percentage of the information about the artefact.

2.2 The population centre

There is only one population centre in the simulation but there are 5 different levels:

```
Hamlet, village, town, city, metropolis.
```

You need to decide the population ranges for each.

Each level should add relevant functionality on top of the previous one. It's up to you to determine what that functionality is.

As an example, a town may have a medical centre. The name of that functionality may change when going to a larger population centre, so it might be called a hospital in a city, but there should also be something distinctive added as well.

In jumping you need to determine if a transition is needed, initially from metropolis to city. Transitions should be managed using copy constructors between adjacent levels. This means if your jump takes you from a metropolis to a town you go via a city.

Usually moving backwards in time results in a population decrease, and forwards a population increase, but the rate should vary depending on the population centre and the technology level. You should likely determine a base growth rate, possibly with saturation points, and have modifiers based on the population centre level, technological era, and events.

2.3 Technological eras

These are different time periods. These could be periods like the Stone Age, Middle Ages, or Information Age. It's up to you to determine what they are called and the impact they have.

Usually moving backwards in time results in a technology decrease, and forwards a technology increase, but the rate should vary depending on the population centre and the technology level. You should likely determine a base growth rate, possibly with saturation points, and have modifiers based on the population centre level, technological era, and events.

2.4 Events

These occur within the year by year step with the chrononauts in the population centre. There could be development level and technology level change rates as a result of chrononaut influence, although they are supposed to limit this.

The following describe the events that could occur. The details are up to you, and there may not be an event every year.

1. Finding artefact information. Assume there is only one lot of information that can be found in a jump site.
2. Finding a jump clue. Assume only one jump clue can be found in a jump site.
3. Very small chance of discovering the artefact you have, "back" in 2525, is actually a fake. End of this search.
4. Illness ... Plague: Decrease population significantly. Could lose a chrononaut.
5. Skirmish ... War: Increase technology, decrease population. Could lose a chrononaut.
6. Technological breakthrough: Increase in technology.
7. Social revolution: Some sort of Decrease or Increase in the population, depending on the nature of the revolution.
8. Interactions between chrononauts and locals: These could be positive or negative, and could change population, technology, ... It's up to you to determine the details.
9. Chrononaut specific event: There should be some specific event relating to the chrononauts that you make up.

To jump you have to obtain both the information and clue. The chances of these appearing should increase year by year, so that both should be found within 10 years.

2.5 The chrononauts

Each chrononaut should have a name, actual age, and travel age. The actual age doesn't change. The travel age is the accumulated years spent in the population centre.

Provide characteristics for the travellers below to support their impact on the events. Each should have an ability level which influences how well they can perfom their task.

The types of chrononauts are as follows:

- **Jump engineer**: If you lose the jump engineer you are stuck. Jump engineers have special brain wiring that interferes with other functionality so they are otherwise pretty much useless, but they are also rare.
- **Doctor**: Determines the likelihood of introducing or suffering from disease/death.
- **Historian**: Determines the value of clues.

- **Security**: Determines impact of interactions with locals.
- **Chrono-Pet**: Should have some impact, to be determined by you.

You expect that abilities will slowly improve, as a function of technology level.

2.6 Wrapping things up

At the end a summary should be given for each artefact. This should include a list of the information elements found, and when they were found; and the overall information percentage found. Major events should also be listed.

Notes on submission

Submission is via Moodle.

Your code must compile on Banshee with the instructions you provide. If it doesn't you will likely be given zero for the programming part of this assignment.

Please submit your source, so .cpp and .h files, Readme.txt file, data files in used, report, and makefile if you have one, in a zip file A2.zip. There shouldn't be any directory structure within the zip file.

Make sure your report is in pdf and has your name, student number, and lab class marked clearly on it. The final diagram should be in your final report pdf and not in a distinct document. It may be ignored if it's a non–pdf format.

Your early diagram submission should also be in pdf, and should also have your name, student number, and lab class on it. You can split it across multiple diagrams if necessary.

1. The early diagram submission is **7pm Friday** 20^{th} **September 2019.**
2. The main submission deadline is **11:55pm Sunday** 13^{th} **October 2019.**
3. Late submissions will be marked with a 25% deduction for each day, including days over the weekend.
4. Submissions more than three days late will not receive marks, unless an extension has been granted.
5. If you need an extension apply through SOLS, if possible **before** the assignment deadline.
6. Academic misconduct is treated seriously. Students involved will likely receive zero.