SIT102 Introduction to Programming

Distinction Task 7.3: Custom Program

Overview

For this task you are going to demonstrate that you can apply what you have learnt to the design and development of your own custom program.

While this is a Distinction task, there are criteria that describe how your program can demonstrate high distinction level outcomes. Achieving this will mean that you are eligible for a High Distinction result.

Submission Details

In this task, submit the following files to OnTrack.

- A brief document outlining what your program does, and the structure of your solution (the data types you created, and the key functions/procedures and how they work together)
- A single concatenated file with all of your code in it (both headers and code files). We want you to work on this in separate files, and just combine them together to submit at the end
- A screenshot of your program running
- A screencast demonstrating your program in action. See notes below.
- A second screencast of you talking though the code (around 2 to 3 minutes). See notes below.

Instructions

You have now completed tasks related to all of the unit learning outcomes, and can work toward demonstrating these in your own program. If you are aiming for a Distinction or higher grade you should start working on this program now. Aim to create something of at least the complexity of the Space Game program.

Here are some steps to get you started:

1. Contact your tutor to get feedback on your program proposal.
2. Start thinking about the data types you need to create to manage this.
3. Think about the functions and procedures you can create to get this working.
4. Write up a short description of how this works.
5. Start coding this using an iterative process: build a little, get it working, then build some more.

Note: Your program should be different from the other programs in the unit. You want to demonstrate that you have learnt from these tasks and can apply what you have learnt to some other program design.

To be eligible for a Distinction your program needs to demonstrate the following:

- Demonstrate the ability to effectively design data types (structs and enums) to model the data

associated with your program
- Demonstrate the use of functional decomposition with use of functions and procedures
- Demonstrate the use of vectors
- Demonstrate the use of different control flow structures: sequence, selection, and repetition
- Demonstrate appropriate use of the C++ coding conventions used in the unit - case, indentation

To be eligible for a high distinction, your program needs to demonstrate the following:

- Your ability to design and implement a program of reasonable complexity.
  - Program does more than have the user respond to random actions, or simply manipulate data.
  - The program must demonstrate the need to think about its structure and implementation.
- Effective use of functional decomposition.
  - Many smaller functions/procedures, where each has a clear purpose/task within the solution.
  - Little code duplication, with code effectively reused.
  - Data is used intelligently to minimise the amount of code required.
- Use of good programming practices.
  - Code is correctly indented, with meaningful names assigned to all identifiers.
  - Code is commented to help the reader understand the abstractions and how they work.

Write up a short report describing how your program works. Describe the core data types, and functions & procedures in your program, and how they help achieve your program goals.

Screencasts

There are 2 screencasts that we want you to produce to help us understand what you have created.

1. **Demonstration screencast**: Use this to show us what your program does. Demonstrate its main features, and anything you think would be good in a “promo” style video.
2. **Code screencast**: In this we want you to walk us through your solution. You have about 2 to 3 minutes to cover off on the main aspects of your program. In this we want you to touch on the following aspects:
   - A quick overview of how your program fits together - imagine we are going to help you extend this program. What would you tell us to help us get started quickly.
   - How did you go about building the program? What problems did you encounter, and how did you overcome these?
   - Highlight any other aspects of interest in your program that are interesting. Use this to wow us with what you have created.

Upload the screencasts somewhere and then paste a link to your video in the task comments. This can be posted to YouTube if you want, or you can use DeakinVideo if you want a private location.