SpaceXplorer

Introduction

"SpaceXplorer" is a text-based game created in C, where players take on the role of an astronaut operating a spaceship, exploring a virtual deep space. Your goal is to return home while exploring space, cleaning up space junk and upgrading or fuelling your spaceship from recycling space junk, while navigating a 2D space world that consists of a grid, with a minimum size of 18x18 cells. The text based element could be descriptive or it could be visualised with laid out letters/characters on screen used to represent the world (but not use “graphics” as we know them).

As an intrepid space explorer, you'll embark on a thrilling quest through deep space, searching for space junk, possible alien encounters or exploring new worlds. To succeed in this programming coursework, your game must meet the following criteria:

SpaceXplorer Requirements:

* Harvest/collect space junk moving through space under your control.
* Fly your spaceship around in deep space with a finite amount and diminishing resource of your choice.
* Traverse deep space while avoiding an asteroid which moves on a fixed trajectory at the same rate as you do (one cell per move you make).
* Minimum “space” size of 18 by 18 cells (there is no maximum but consider the playability of your game)
* Have a win and loss condition

Game Mechanics:

* The placement of space junk is randomized at the beginning of each game, offering a unique experience every time.
* Each cell in the world can contain only one piece of space junk at a time.
* Your character, the bold but lonely space traveller, is under your control and moves one cell at a time in the up, down, left and right directions (but not diagonally).
* Ensure that your spacecraft remains within the boundaries of deep space (the 18 by 18 minimum grid).
* The player has a limited resource (perhaps air, fuel or time? You choose) that slowly decreases/runs out but that can be refreshed according to a game mechanic of your choice, and the player will lose the game should your character run out of this resource.
* A spaceship health monitoring system, perhaps you need to harvest space junk, recycle it and repair or upgrade your ship? The player should be able to check on the spaceship health on demand.
* The character should be able to somehow use the space junk – you choose how or what they do with it.
* If the moving asteroid comes into contact with the spaceship, then it is game over.
* The game is won when the character meets a specific threshold of your choosing (points? Cleaning up space? The possibilities are endless).
* The game’s difficulty should be chosen by the user and should feature an element of scalable difficulty of your choosing.
* Read some data from a text file (i.e. Level settings, an introduction text?)
* Write some data to a text file (i.e. Leaderboard of scores?)
* There should be a loss condition, that is triggered by running out of the limited resource and also by hitting the moving asteroid (and any other things you like – alien encounters perhaps?).

Programming requirements:

* Appropriately use Structures
* Appropriately use Pointers
* Use custom (your own) header file(s)

**Assessment of the Mini Coursework:**

Part A: Development log of evidence containing the following (Due 02/05/2024, with 48hr late period):

* A Structure Chart showing information flow for your game functions
* Pseudocode/flow chart for the overall programme
* Pseudocode/flow chart for each function
* Description of each function, it’s input and output data types and ranges and error handling.
* Version history and change log
* Test cases for unit and function tests, and integration tests (use test case template)
* Test scenarios for game features (use test scenarios, by appropriately modifying the test case template)
* Your full code (all .C and .H files you have produced) – **If you do not submit this code you will get 0 for part A and B**

Part B: Demonstration and Viva

This will be scheduled in the exam period and you will be given in a 30 minute window to demonstrate and explain your code. This will be recorded for marking, moderation and external examiner processes. You will be assessed in the following way (times are approximate):

Demonstrate which features your game has (which of the game requirements can you demonstrate) (approx. 10 minutes), you will be asked to show these in your code and game.

For a marks breakdown and weighting, see Feedback for Mini Coursework.xls file.

Approx. 20 minutes:

You will be asked 3 questions (though there may be follow up questions) about any of the features you have implemented on the demo list above, this is to test your authorship (that you wrote the code), that you understand what you have written and to judge your design choices. You will be expected to have a working knowledge of your code and game, and be able to explain how certain sections of the code work when asked, and will be expected to justify your design choice (why you wrote it that way and if this was a good choice).

Your responses will be judged based on these factors, and marks provided for these questions. Remember, you wrote this code, so have confidence in what you wrote, and why you wrote it, and ensure that if you used resources to help inform your code, that you understand what the code does and how it works. This is not here to catch you out, but instead to ensure you wrote the code and can make good programming design choices.

Please note that should the markers not feel confidence of your authorship, this will lead to an assessment offence investigation in accordance with the UWE Academic Conduct Policy (formerly known as Assessment Offence Policy). <https://www.uwe.ac.uk/study/academic-information/assessments/academic-integrity>

This will be scheduled during the exam weeks and you will be expected to attend your Viva and Demonstration online. Not attending this will be considered as a Non-submission like any exam. As these are online, you are required to have a microphone, webcam and the ability to share your screen, you must also be on your own with no one else in the room as this is an individual assessment – you are responsible for these and failure to adhere to this will result in a non-submission. These will be recorded for moderation purposes.

**Use of Generative AI is not permitted for this coursework**

See here for UWE guide <https://www.uwe.ac.uk/study/study-support/study-skills/generative-ai-study-skills-guide>