**CS225 Semester Project (Rev 0)**

Project Description:

You are to deliver a relatively sophisticated software project plus associated design and test documentation. The software project and deliverables shall have the following properties:

1. The project will utilize a GUI to allow for user input and to display results. All user interactions must occur via GUI widgets and/or GUI menus.
2. The project will utilize file I/O. The specific usage of the file I/O is left to the student, but must include input and output. Typical uses may include saving results, initializing the program, or allowing users to save state and resume later.
3. The project will utilize at least one case of exception handling (in addition to that needed for file I/O) to recover from a condition that would otherwise stop program execution.
4. The project shall include multiple objects and include at least one case of inheritance (in addition to inheritance required for the GUI).

Deliverables: The project has seven deliverables as outlined below. Each deliverable will be posted as a separate assignment on Canvas. The deliverables and schedule are designed to encourage you to follow a process of conception, establishing functional goals and requirements, high level design, and iterative implementation cycles. Deliverable description and grading rubrics are provided below.

* P1 Proposal: The project proposal is intended to get you to conceptualize your completed project before starting on it. You are to provide a brief description of your project, and describe how the project will meet the four major requirements listed above. Fill out the proposal form provided in this document and turn in online.

|  |  |  |
| --- | --- | --- |
| **P1: Project Proposal (10 pts)** | **Points** | **Awarded** |
| Project description | 5 |  |
| Meeting major requirements | 5 |  |

* P2 Project Description and Background: This deliverable provides a more complete description of your project than was presented in the proposal, along with background information, and specific problems that require solution to complete the project. The background description is intended to provide context for the project. If you are creating a game, describe the type of game and previous examples of it. If you are creating a simulation or software tool, describe who might use it and why. Lastly, describe specific computational problems that you need to overcome to complete the project. Do you need to design a file format to save and restore system state? Do you need to compute collisions between objects, or create real time animations? Do you need to perform statistical summaries of thousands of events? These are all computational problems that require thought (and solution) before you can begin implementing the project. Describing them in this early deliverable is intended to start you thinking about them.

This deliverable must be formatted using the report template provided separately. You may adapt the template to suit your project needs.

|  |  |  |
| --- | --- | --- |
| **P2: Problem Description and Req’ts (30 pts)** | **Points** | **Awarded** |
| Completeness of description/background | 20 |  |
| Grammar | 10 |  |

* P3 Functional Description: This deliverable contains a functional description of your project in the form of User Stories. User stories were introduced in the Agile software development process and take the form “As a \_\_\_\_\_\_\_ I want to \_\_\_\_\_\_\_\_.” They are used to describe the completed software in terms of functionality, but are not testable requirements. Writing user stories accomplishes two main things: it helps you to envision what the completed work will look and act like, and it creates a backlog of tasks to be completed that helps you to manage your schedule. For example if you create twelve user stories, you might have a goal of completing two of them every week. Then each week you have to prioritize which two can be completed.

Some examples for a Space Invader – like video game:

* + As a player I want to be able to move my icon left and right.
  + As a player I want to be able to shoot at the aliens.
  + As a game designer I want to be able to move my aliens in a fixed pattern.
  + As a game designer I want to be able to provide multiple levels to the player.

This deliverable must be formatted using the report template provided separately. You may adapt the template to suit your project needs.

|  |  |  |
| --- | --- | --- |
| **P3: Functional Description (20 pts)** | **Points** | **Awarded** |
| User stories (at least ten required) | 20 |  |

* P4 Design and Test Cases: In this deliverable you will provide the design for your project in the form of discussion, algorithms, UML diagrams, and state the tests cases consistent with your requirements. Test cases will be graded for three requirements, though you may include more without affecting your grade.

This deliverable must be formatted using the report template provided separately. You may adapt the template to suit your project needs.

|  |  |  |
| --- | --- | --- |
| **P4: Design and Test Cases (40 pts)** | **Points** | **Awarded** |
| Algorithmic descriptions | 15 |  |
| UML Class Diagrams | 13 |  |
| Test Cases | 12 |  |

* P5 Preliminary Delivery: This is a preliminary delivery of your code only, intended to encourage you to perform coding well before the final delivery date. No new material to your report is required for this deliverable, though your software will be compared against the UML diagram. You must have a class file for every class shown on your UML diagram, even though the class file is not complete. Typically, many methods will be incomplete at this point. You must have at least two User Stories operating correctly within your code for this deliverable.

|  |  |  |
| --- | --- | --- |
| **P5: Preliminary Delivery (30 pts)** | **Points** | **Awarded** |
| In-Code Documentation | 10 |  |
| User stories operating correctly | 20 |  |

* P6 Preliminary Delivery: This is a preliminary delivery of your code only, intended to encourage you to perform coding well before the final delivery date. The requirements are the same as the P5 delivery, except you must have at least six User Stories completed.

|  |  |  |
| --- | --- | --- |
| **P6: Preliminary Delivery (40 pts)** | **Points** | **Awarded** |
| In-Code Documentation | 10 |  |
| User stories operating correctly | 30 |  |

* P7 Final Delivery: This is the final delivery of your code and accompanying report. You report should contain all the elements listed in the grading rubric.

|  |  |  |
| --- | --- | --- |
| **P7: Final Delivery (130 pts)** | **Points** | **Awarded** |
| Title Page | 2 |  |
| Problem Discussion | 5 |  |
| Requirements | 5 |  |
| UML Class Diagrams | 5 |  |
| Solution Discussion with algorithms | 5 |  |
| Actual Time, Lines of Code | 3 |  |
| Test Cases and Results | 15 |  |
| Source Code |  |  |
| *Documentation and Style* | 10 |  |
| *Correctness* | 40 |  |
| Project Requirements |  |  |
| *GUI User Input and Display* | 10 |  |
| *File I/O* | 10 |  |
| *Exception Handling* | 10 |  |
| *Multiple Objects with Inheritance* | 10 |  |

**CS 225 P1 Project Proposal Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |
| --- | --- |
| **Provide an overall description of the project.** | |
| To be used for the Dungeons and Dragons live action role playing game. It is a tool to help new, and returning, players create new characters, as there is a lot of rules to how one can be created, and hundreds of different combinations of character options. This tool can be used to quickly and much more easily create a character to be used in the game. Eventually will be able to take into account character levelling and will update stats and inventory as stuff is added, this last part may not make it into the final project. | |
| **Describe how the project will implement the following required features.** | |
| **GUI** | This will be how the user chooses the character that they want, will have descriptions of each race, sub-race, and class to help the user pick which one best suits them and will create a printable character sheet. |
| **File I/O** | Will create a file that holds all of the character data, and will write to it and read from it as the user makes decisions on the character they want. Will eventually have the option to print out this file. |
| **Try-Throw-Catch** | Prevent the user from being able to choose multiple races or starting classes |
| **Inheritance** | Races and classes in general have basic things they can do, with each specific race/class just a variation. The different races and classes with inherit the characteristics of a generic race or player. In addition, there are sub-races such as half-elves that are elvish, except for a few differences meaning they would inherit the elf race class and then add a little bit more to it. |

|  |  |  |
| --- | --- | --- |
| **Project Proposal (10 pts)** | **Points** | **Awarded** |
| Project description | 5 |  |
| Meeting major requirements | 5 |  |