

GDDI 590.01 Final Project

Due Date: 12.5.2025

Total Points: 100

Extra Credit: 5 pts each (up to a max of 120)

Overview

The final project will be a command line game of your own design.

All programming must be done using C++.

Minimum Requirements

- Your project MUST include a readme, which should include the following:
 - o Your game's title
 - o A backstory for your game that introduces the world and protagonist (if applicable)
 - o Your game's primary goal(s)
 - o Your game's primary game feature(s)
 - o Gameplay instructions
 - o List of [additional C++ features](#) implemented by your game
- Your game MUST be able to load and save a file
 - o This could be game state, gameplay instructions, NPC text lines, etc.
- Your game MUST support player interaction using the command line
- Your project MUST compile successfully using a Makefile
 - o **EXTRA CREDIT: Cross-Platform Compatibility – can you figure out how to support compilation to Windows, Linux and/or OSX?**
- Your project MUST NOT leak memory
- Your project MUST use the standard library
- Your project MAY use the following 3rd party libs:
 - o [Raylib](#) (raudio only – if you want to include music)
 - o [ncurses](#) (text-based UI framework)
 - o [nlohmann](#) (JSON)
- Your project MUST demonstrate the following C++ language features:
 - o C++ Classes

- Constructor Overloading
- Operator Overloading
- Inheritance
- Encapsulation
- Polymorphism (either `static` or `dynamic`)
- Dynamic Memory Management (`new` and `delete`)
- Error Checking using `static_assert` and/or `assert`

Additional Coding Requirements

In addition to the above minimum requirements, your game should implement additional features selected from the list below.

Implement *at least* 3 of the following:

- Linked List (`single` or `double`)
- Binary Search Tree
- Abstract Class
- Templates
- Dynamic Polymorphism with Virtual Functions
- Recursive Function
- Hash Table
- ASCII Graphics
- Load and Save Game
- Splash Screen
- Unit Tests
- Compile and Link Your Own Game Library
 - This would involve organizing your code and build process such that some of your files are compiled into a separate binary which is linked to the primary game executable

EXTRA CREDIT: You may implement **MORE** than 3 of the above features. Each additional feature is worth 5 points (to a max of 120).