



Assignment 4 Requirements

[Assignment 4 Discussion Forum](#)

Assignment 4

Game Project

Weight: 25% of your final grade

▼ Instructions

Refer to the Study Guide as needed to ensure you have the understanding of the course material required to complete this assignment.

Refer to the [Important Information about Programming](#) for the most up-to-date software (code editors and compilers) for C++ programming.

Use the Assignment Drop Box on the course home page to submit your assessment. Follow exactly the format for submission outlined in the [Assignment Requirements](#).

All assignment submissions must include all of the following deliverables:

- C++ source code for a complete, working program that fulfills the requirements of the following project;
- additional files as described in the requirements for the project;
- full documentation and a test plan for your primary C++ source code file as described in the Assignment Requirements; and
- reflections on the assignment (Assignment Journal) as described in the Assignment Requirements.

NOTE

A significant number of marks is attributed to your documentation and test plans. Be sure to include all deliverables with your assignment.

▼ Overview

This is a programming project. It represents a significant amount of programming work and a substantial investment of your time. You will need to research programs using Internet resources and the discussion forum for this assignment, and then design your program to include the major Elements of the User Experience (described below). Moreover, you will need to code your program, document it, and test it.

What you will be researching, designing, building, documenting and testing is an adventure game. Text-based adventure games were common and popular in the 1970s, 80s and 90s.

The grandfather of all text-based adventure games is Colossal Cave Adventure, which also went by the names *Adventure* and *ADVENT* (on early systems with file name limits). *Adventure* was created by Will Crowther and Don Woods in 1975–1977 and originally written in FORTRAN (a third-generation computer language) for PDP minicomputers. It has been ported to many other systems and rewritten in several other computer languages.

Your first line of research is to find out more about this game, especially the style of game play and how the game was constructed. An Internet search on topics such as “Adventure (game)” or “Colossal Cave Adventure” will turn up plenty of information. A good starting point is [Wikipedia](#).

Ultimately, your game will comprise all the following elements, and will form a complete, playable game.

▼ Elements of the User Experience

The game, itself, is a *text-based adventure game*. There are no graphics; only text is used for communication between the game and the player. All output is descriptive, in the form of paragraphs and short answers (e.g., “You can’t go in that direction.”). All input is in the form of simple text commands chosen from a limited set of options (e.g., “go north”). The input is sophisticated, though, as it recognizes “go north,” “north,” and “n”

as identical commands, and acts accordingly. Punctuation and capitalization are ignored as input.

You are the main character or player. The game space is represented as a set of rooms, connections between rooms, characters you meet, and items. Items include things you will need to successfully complete the game, such as food, rope, flashlight, etc., as well as items you find that will increase your final score, such as treasure, rugs, coins, batteries, etc. Items found can be carried, dropped or used. Carried items are maintained in an inventory accessible with the "inventory," "invent" or "i" command.

Items can be combined with action words to cause action events in the game (e.g., "throw rock at goblin").

Sometimes action event outcomes are random. For example, if you throw a rock at a goblin, the goblin might duck or it might get hit.

The player might encounter other characters, such as a thief, goblin, bird, snake, etc. Some are good characters, and many are evil. Encounters are random and occur at any time or location. Encounters may be triggered by an item in the inventory (e.g., if you have a jewel, you are more likely to encounter the thief).

The player begins the game at a common location, and proceeds to explore the game world using the input commands and responding to the displayed output. Some rooms have multiple entrances and exits, and some have many ways in, but only one way out (e.g., sliding down a tunnel). Sometimes you may encounter a maze, and dropping an inventory item will help you navigate out.

The play proceeds with the player inputting a text command sequence. The game then responds by displaying the outcome. Sometimes the outcome will be a long, descriptive paragraph explaining what is happening. Other times, it will be a description of what the player can see at that location. Usually, the first time visiting a location will trigger a long descriptive output, and subsequent visits will display an abbreviated output (e.g., "You are at the chasm again").

The game proceeds until the player either perishes or wins. The player can perish in an interaction with a nasty character, by starvation or thirst, or by

falling down a hole or off a cliff. A win is usually indicated by a return to a "safe area" with all the treasures.

The establishment of a safe area can benefit the game play by allowing the player to "stash" inventory items for safe keeping or to relieve him or herself of carrying multiple items.

▼ Your Game

The game you write for this assignment is based on the book *Alice's Adventures in Wonderland* by Lewis Carroll, also known as *Alice in Wonderland*. You may base your game on either the book by Lewis Carroll or one of the movie adaptations of the book.

Alice's Adventures in Wonderland is in the public domain. You can access a copy of the work online through [Project Gutenberg](#).

Alice in Wonderland makes for an entertaining adventure game, because there are many rooms (scenes), as well as a number of key characters and situations you may employ in your game.

You are not expected to follow the entire plot of the book, nor are you expected to program the entire book, all the items, or all the characters.

Rather, your game is to be an adaptation of the book into the form of an adventure game in which your character—Alice, perhaps—sees various scenes and characters and can interact with them and with the items found in their vicinity. There is no requirement that your main character (i.e., you, the player) represents Alice, although this is probably the most common choice.

For example, you may start your game as Alice, above ground, who encounters the white rabbit and talks to him. Eventually, you find the rabbit hole and fall down to the room containing "eat me" and "drink me." Consuming these items will presumably have the same effect as in the book.

Likewise, you may interact with the Cheshire Cat, the Mad Hatter and other book characters. Prolonged interaction with the Red Queen will, presumably, prove to be unfortunate.

Ultimately, you will decide how the book characters, scenes and items "come to life" in your game.

▼ Project Requirements

The game must employ object-oriented principals for all the components. That means the highest class is *Game*—your finished game. You may choose to call this main class *Wonderland* or another appropriate name. The Game class should have one main() method that creates an instance of the game and invokes it for game execution. Within Game, necessary classes must include Locations, Items, Characters, Actions, Inventory and Control. You may find more classes are necessary to complete the game.

The Control class is the class that reads user input and checks the input against valid Actions. If an Action is allowed, then the methods required to complete the action should be called on the objects involved (e.g., Action: "take"; object involved: "flashlight"). Certain actions will result in changes in the user's inventory. Other actions will change the user's location. Changes in location will trigger appropriate descriptive outputs based on the room entered and whether it is the first time entering.

Give careful consideration to your how the objects (classes) in your game interact.

Your game can contain any reasonable number of locations (rooms), with a minimum of five (5). A common starting point is near the rabbit hole, and the ending point is somewhere in wonderland. Once you descend into the rabbit hole, you are unable to return. Otherwise, the locations (rooms) and how you connect them is your design choice, which MUST be completely documented as part of the design documents for the game. You should create a "safe room" to act as a gathering area for your treasures.

Locations should have a minimum of one entrance and one exit (not the same), but at least one room must have more than three (3) exits.

You must portray a minimum of five (5) characters from the book. You must have a minimum of 12 items for the main character to interact with, which should be taken from the book.

You will need to establish a goal for the main character (e.g., Alice), and you must state this at the beginning of the game.

The items, characters, location descriptions and action words should be contained in separate text files, which are opened and loaded into the game at the start of the game.

▼ Deliverables

In addition to the standard deliverables described in the Assignment Requirements (located under Assessment on the course home page), your final submission for this assignment should also contain:

- The game text files (items, characters, place descriptions, actions)
- A design document in which you fully describe your game's design elements, concept, plan, play and goals. This document should also explain all the classes and text files used by the game. This document may be written in point form or full sentences, and it should be 1000–2000 words long.
- A document in which you discuss the research you conducted to create the game, as well as:
 - your design decisions (that is, your reflections on how you arrived at the finished design);
 - the challenges you faced and how you overcame them; and
 - the testing you did and how you solved any issues found in testing.

This document should be no less than 1000 words in length, and no more than 5000 words. You may incorporate this portion of the project into your Assignment Journal, if you prefer.