Adventure Game Project

High-Level Design Plan

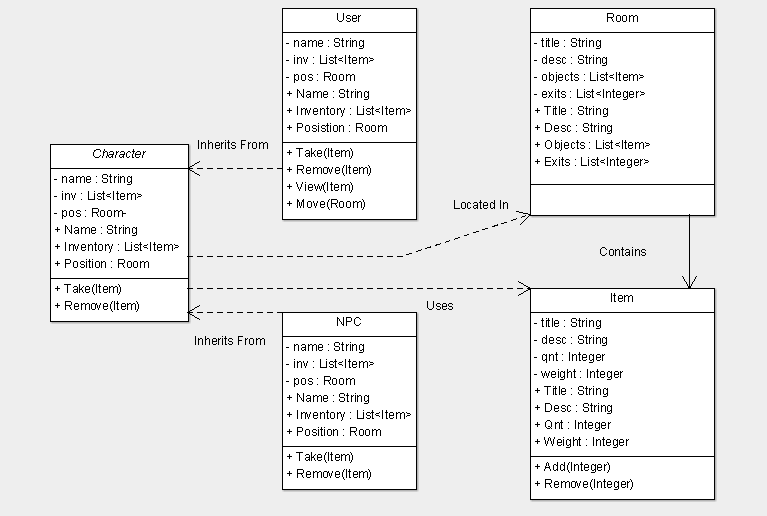


Group: Brent Rubell, David Austin, Eric Pires

Requirements Specification

Requirements%20Specificiation.png

UML Class Diagrams



The UML diagram contains all the classes used in our Main class. The *Room* class contains Item classes and can call the methods inside Item to add or remove a specified quantity of items when a character interacts with their inventory. The *Character* class is a parent to both User and NPC. These two subclasses contain all attributes and share two methods. The *View()* and *Move()* methods are not shared. These methods are used only by the player; therefore, they are omitted from the parent and assigned specifically to the child class *User*. All low-level functionality will be implemented inside functions called in the Main class. These functions are not included in this high-level diagram; however, they still exist in our program.

Program Flow Description

ProgramFlow.png

Planned division of Work

The group unanimously agreed to use the Microsoft C# specifications and coding style, when working on this project. Having consistency was an important aspect everyone agreed on, and by maintaining the same style, that will be able to be accomplished. Along with the same specifications and coding style, all members will also follow the same comment style which is:

/\*\_Comment\_\*/

Console.Write(““);

Each line of code will be limited to 16 characters per line, in order to maintain readability among users.

All members will contribute and integrate player interaction with the game, because it is an aspect that relates to each person’s piece. As the project progresses, all members will optimize the current code, in order to improve the code as much as possible. Along with the coding responsibilities, all of the group members will also split up the technical report, user guide, as well as showing a demo of the game to the class. In order to catch bugs and glitches, an external person will be used in order to test the game, and help improve the program as a whole.

Brent: Brent’s portion of the work will be templating the aspects of the project, such as the start screen, as well as building the diagrams on the computer. He will also be responsible for importation of the XML file, in order to load a map, along with rooms, items, an adventurer, NPC’s. Brent will also be working on implementing testing automation in order to test various versions and conditions for the gameplay, and to catch any issues that come up with the program.

Dave: Dave’s portion is to implement the Rooms and Items, and that will be modular in order to work alongside the XML importation Brent is designing, along with the aspects performed in Eric’s portion. A “Room” class will be created, and it will contain a list of rooms that are imported from the XML, and it will populate with elements such as short description, long description, objects, location #, and exits. The class will be tested in order to verify that it performs as will be required once the other pieces of the project are implemented. Dave will work alongside Eric for the implementation of Items, although both will be working on different aspects of Items.

Eric: Eric is responsible for the character classes, which involves dealing with the Adventurer, along with the NPC’s that will be used in the game. Eric will work on the aspects of character, such as the inventory, picking up and dropping items, along with working in the game environment. He will be working on the dialogue that will be used in game, along with the outline of the maps containing the rooms and items for the game.

All members will work together in order to perform their specified division of work, and will be able to provide support for issues that come up with each other’s aspect of the project. The group will meet up and work together throughout the timeframe of the project in order to get the implementation across all aspects correct.