# Tutorial 7: Intro to Assignment 2

## Class Diagram

Consider the following scenario (this scenario will form the basis of your assignment 1):

Create a game prototype using modern C++. This should be a text based game. The game is laid out as a map. The map is composed of areas which have at least one entrance and exit. Each area is composed of rooms. Each room may contain some items to be picked up and stored in a character's inventory. This game map should be represented by a stl container of shared_ptr to area objects. The various objects that are part of the game map will have links between them represented by pointers between the objects. All containers that are part of your design should be containers of c++ safe pointer objects.

A character has various attributes (ints) that determine their effectiveness in battle (no requirement to implement actual battle but you do need to have values recorded for these stats. The stats include (all will have a value between 5 and 25):

- Int: the strength of the player's magic
- Wis: the strength of the player's healing powers
- Str: the higher the value, the more impact of each blow
- Dex: the larger the value, higher the probability that the player can dodge the blows
- Con: the larger the value, the lower the impact of each blow that lands

Each character will also have status points (each game "tick" (one second in the game) the count of these is increased based on the player's level - the formula will be provided in the final specification):

- Hit points: the health value of the player
- Mana: the amount of magic that can be cast: each spell will deduct from this amount
- Moves: the number of moves the character can make before they are exhausted.

You should provide a login screen for players to log into the game which will drop the player into midgard temple which is the default starting point for each player session. The player must be able to interact with their environment in the following ways:

- "look" without any parameters: display the name and description of the current room, its contents and the available exits (if the door to that other room is closed, the exit will not be displayed.

- "look <direction>": look in the direction specified and see the name of the room there or a message "door is closed" if the door is closed.

- open <direction>: open the door in the direction specified.

- Close <direction>: close the door in the direction specified.

- “look item” where “item” is the name of the item to be looked at: name and description of the item as well as the type of item. For example:

  Name: Shining Sword
  Type: weapon
  Description: A shining sword which will blind the opponent with every hit.
  Level: 1

  Name: Helm of the Gods
  Type: helmet
  Description: this helmet protects the wearer from any unnatural curses.

- “wear item”: wear an item in a wear location where “item” is the name of the item. The item must currently be in the player’s inventory at the time the command is issued.

- north, south, east, west: move through the exit in the direction specified.

- list: list the items for sale - must be in the presence of a shopkeeper

- buy <item>: buy the item specified by name or number from the list of items for sale: must be in the presence of a shopkeeper.

- Account: display the number of gold coins the player has on hand.

- quit: quit the game

Finally, the user of your program must be able to specify at startup the format and location of the data files (the directory). The format of the files may be binary or ascii. Example data files in both ascii and binary formats will be provided to you.

# Lab this week:

Start implementation of the game map for this program.