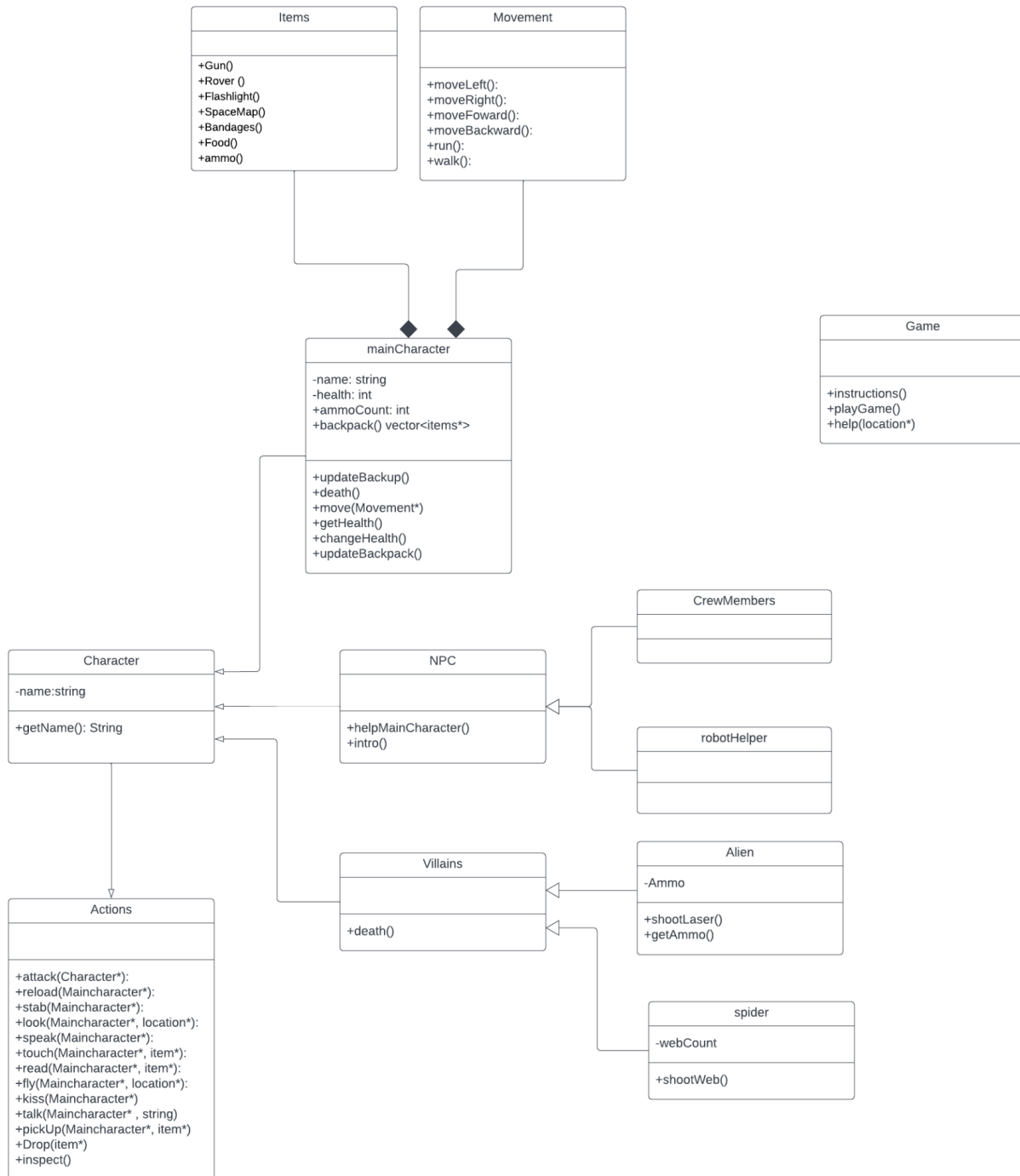
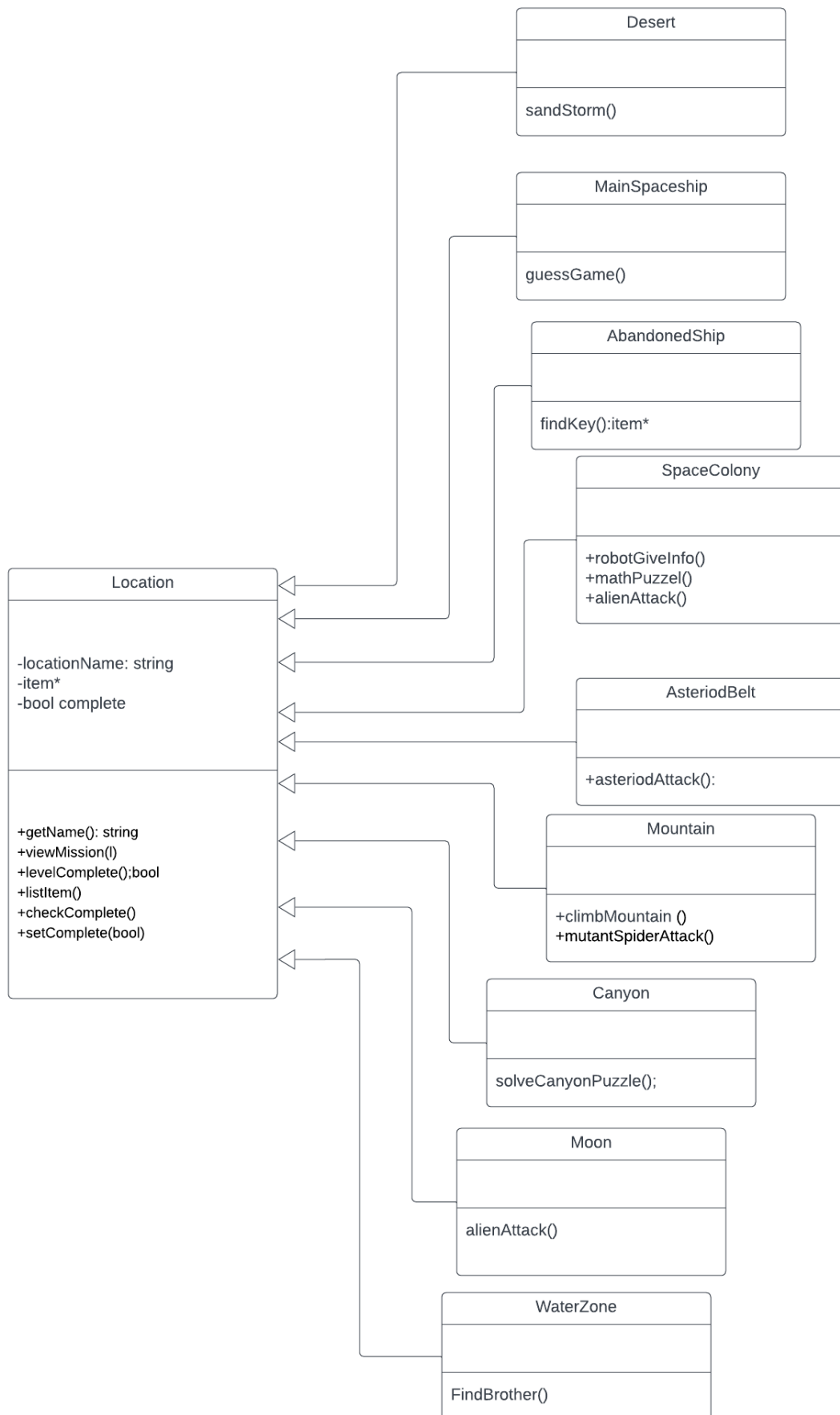


# UML and Class descriptions





## DESCRIPTION OF CLASSES

### **Class:**

**1. Character-** Implements the common features between all of the subsequent sub-classes including the name and the getName() method.

- a. **MainCharacter-** this class implements our main character and includes basic things related to it such as name, health, ammo count, along with the methods for a death and a backpack.
- b. **NPC-** in this class we will implement our NPCs, they will be available at different spots throughout the game and can provide some help and advice to the main character. It will include methods helpMainCharacter and intro
  - i. **Crew Member-** This class will implement some form of the virtual methods helpMainCharacter and info that are defined in the **NPC** class
  - ii. **Robot helper-** This class will implement some form of the virtual methods helpMainCharacter and info that are defined in the **NPC** class
- c. **Villains -** this class will be used to implement functionality of the opponents of the main character. Mainly some aliens that our main character will fight with on his way to complete the game.

**2. Movement-** This class directly associates with the mainCharacter class and has functions for the movement of the main character.

- 1. The movement examples include: moveLeft(), moveRight(), moveForward(), moveBackward(), run(), walk().

**3. Items-** items class have methods for all the items available in the game. The methods of this class will include all the things that character can do with these items and the effects they will have on it.

- 1. Items include a gun, rover, flashlight, bandages, food, and ammo.

**4. Actions-** this class is responsible for all the actions that our main character can perform. The methods of the action class will take a pointer to the main character and will be responsible for changes in attributes of mainCharacter class.

- 1. Actions include: attack, reload, stab, look, speak, touch, read, fly, kiss, talk, pickUp, Drop, and inspect.

**5. Locations-** The Locations class will represent the different places the main character can travel. In this base class, each location will have a name, along with a method to getName(). It will also have a viewMission() method that will be pure virtual, and display the mission of the respected location. Each Location will also have a bool variable that is initially set to be false, and once the level is passed, it is set to true. As a result, a checkComplete() and setComplete()

method will also be pure virtual. Locations will also have one item for the character to interact with.

- a. **MainSpaceShip**- This location subclass. It will be considered the main base, and the user must return here every level. To begin, they must solve a puzzle where they have to guess a word. Then they can choose to access all levels, except the last.
- b. **Desert** - This class is a location subclass. It contains one method, which is a sandStorm(). In this location, the user will try to navigate through a sandstorm. This location has a possibility to end the game.
- c. **AbandonedShip** - All this subclass contains is a key in which the user will be able to use in a future level.
- d. **SpaceColony** - This subclass will contain a robot that gives you info on how to beat the aliens, but first, you must solve a math puzzle. Then the aliens attack you.
- e. **AsteroidBelt** - This subclass contains an asteroidAttack() method, in which the user attempts to dodge asteroids, with the possibility of death.
- f. **Mountain** - In this subclass, the user will have to navigate up a mountain, and when at the top, they have to face a mutant space spider.
- g. **Canyon** - This subclass will contain just a puzzle that the user must beat in order to complete this location.
- h. **Moon** - In this subclass, the user will fight a larger, harder to kill group of aliens than in SpaceColony() class.
- i. **WaterZone** - This is the final location that is only accessible when the user has completed all the other levels. Then, the player will find his brother, and complete the game.