# Zork++

## Phase 1

### Use Case

### UML Diagrams

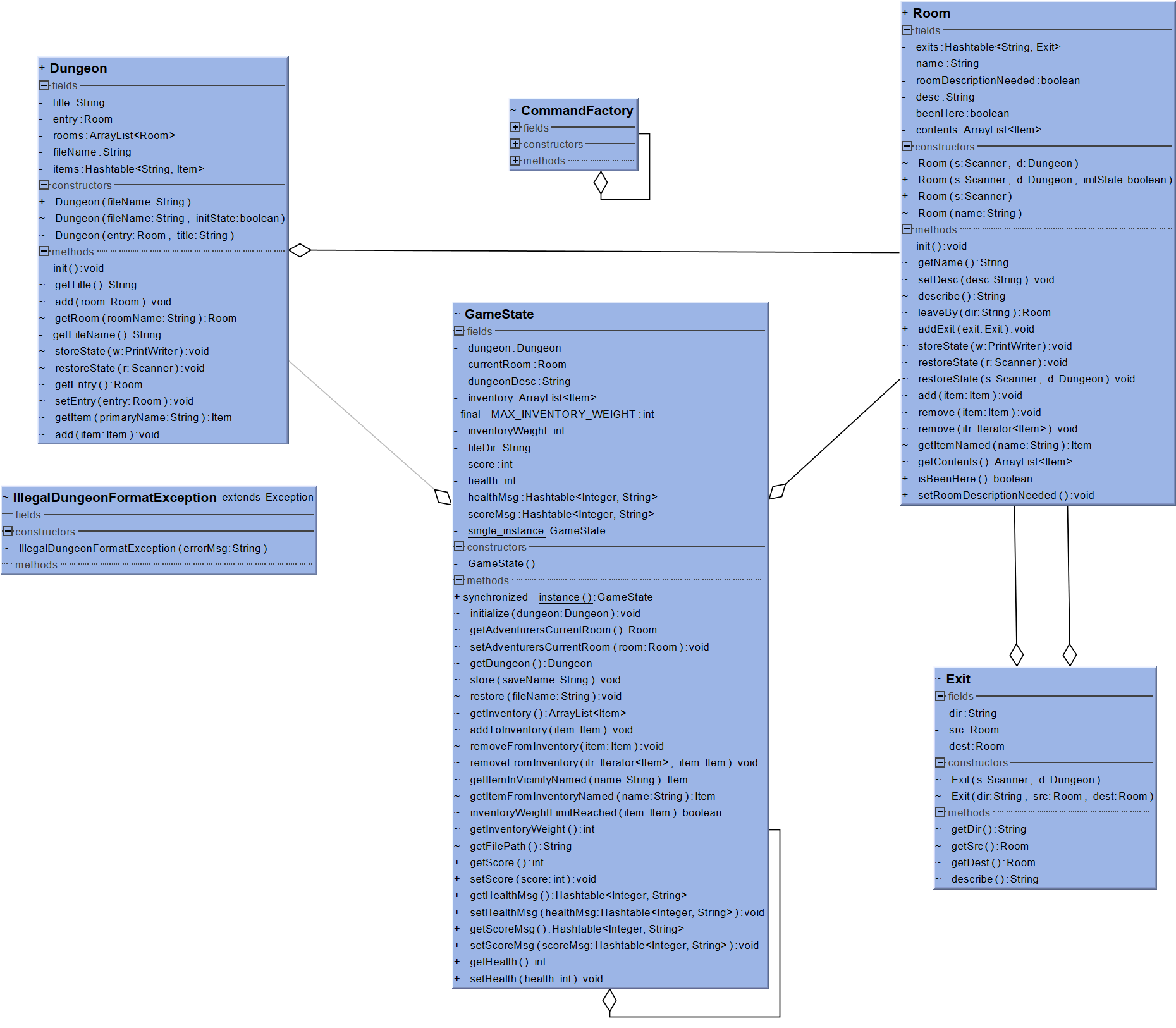


Figure 1: Zork++ Main classes diagram

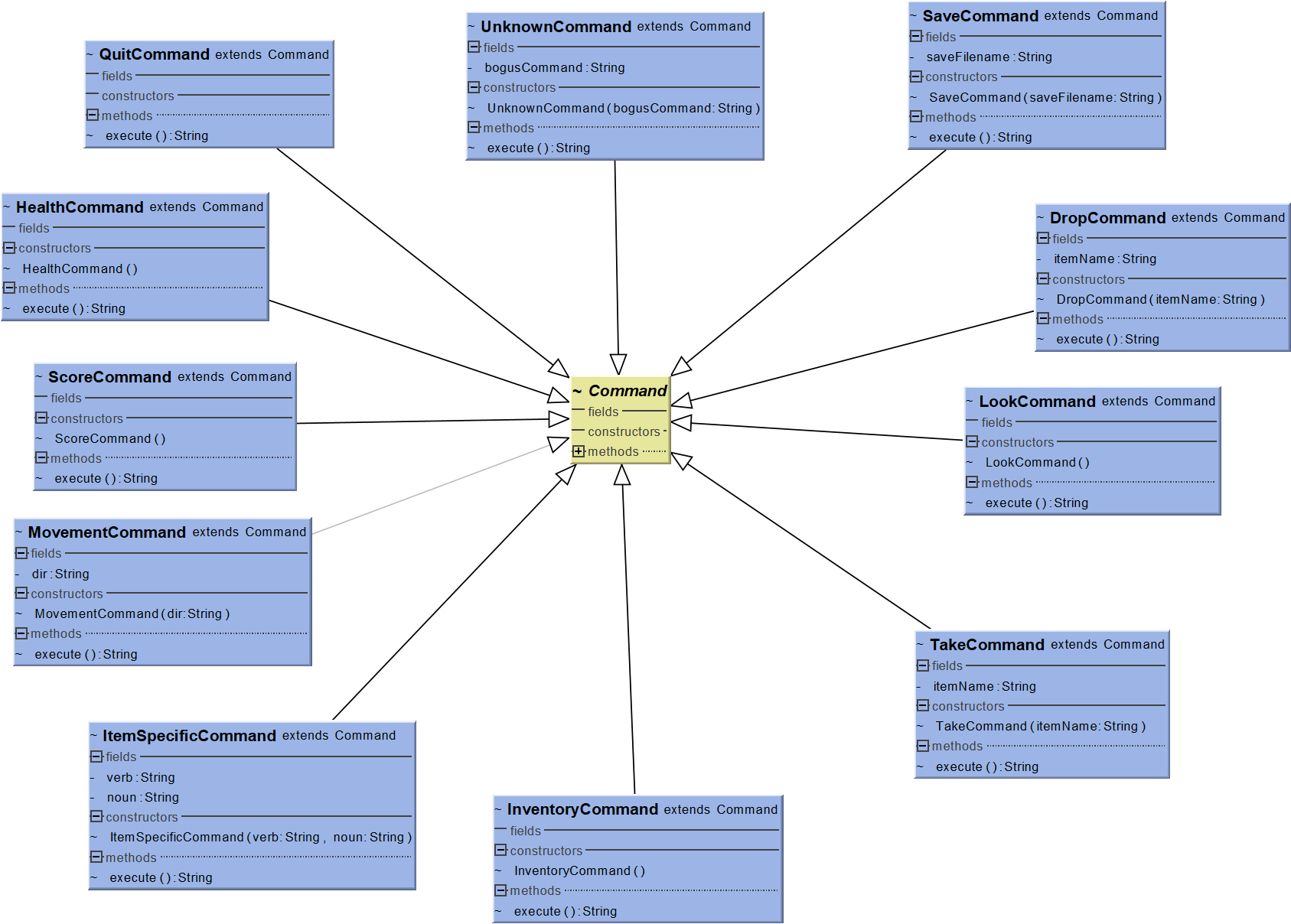


Figure 2: Zork++ Command classes diagram

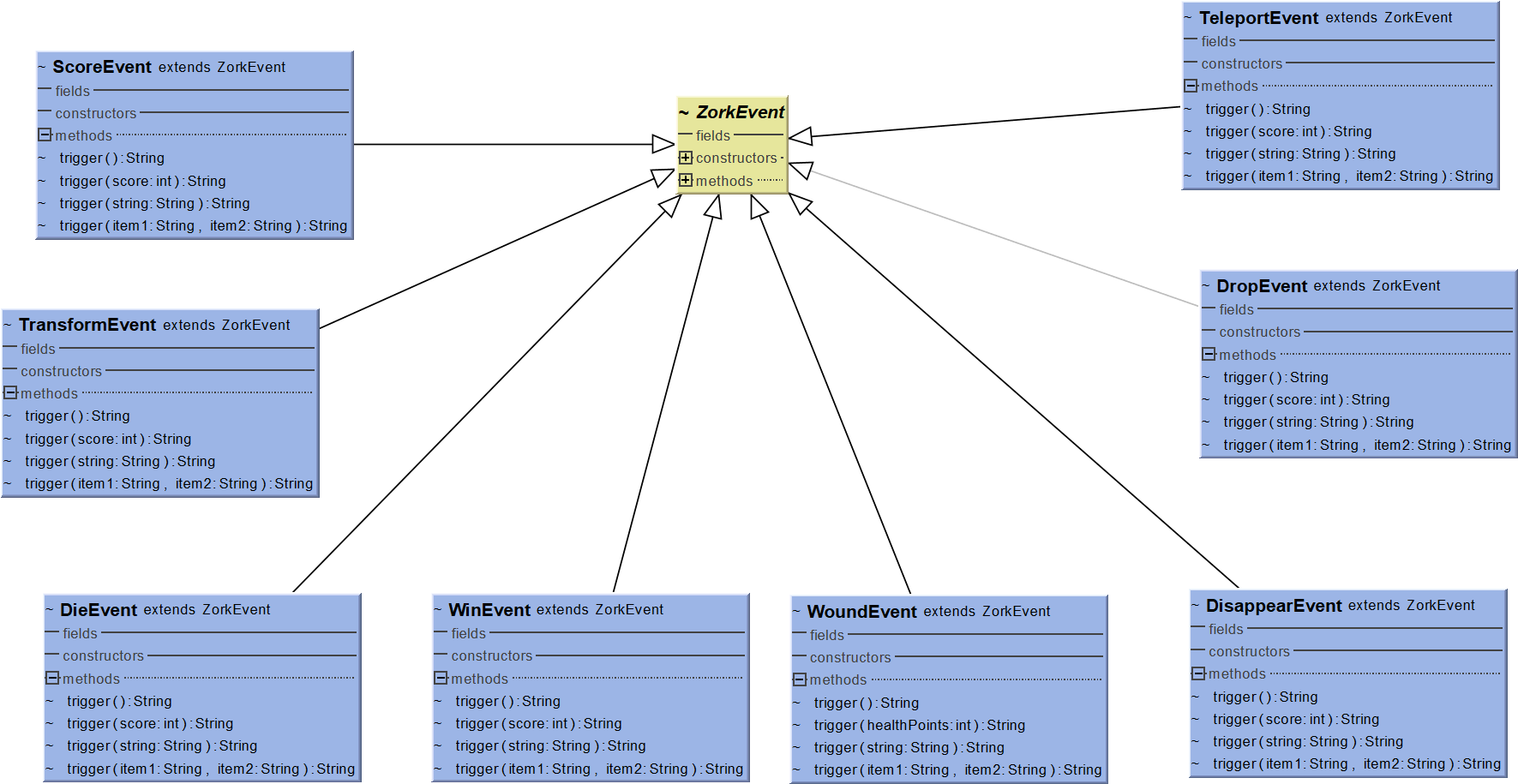


Figure 3: Zork++ ZorkEvent classes diagram

### Sequence Diagrams

### Canonical Examples

### Zork File Format Description

Line 1 – Dungeon name

Line 2 – Zork version (Zork++)

Line 3 – === delimiter

Line 4 – key word (either Items, Rooms, or Exits), followed by a colon (:)

Key word Items, followed by a colon (:)

Multiple items until delimiter (===)

* Each item consists of:
  + Primary name and optional aliases separated by a comma
  + Item weight (int)
  + One or many optional commands, optional events (in square brackets), optional event parameter (in parentheses), delimiter (:), message
  + Delimiter (---)

Key word Rooms, followed by a colon (:)

Multiple rooms until delimiter (===)

* Each room consists of:
  + Room name
  + Room contents (Key word Contents: items inside that room, separated by a comma), if applicable
  + Room description (multiple lines)
  + Delimiter (---)

Key word Exits, followed by a colon (:)

Multiple exits until delimiter (===)

* Each exit consists of:
  + Current room
  + Valid direction
  + Room destination
  + Delimiter (---)

### Save File Format Description

Line 1 – Zork version save data

Line 2 – Dungeon file name, preceded by “Dungeon file” and a colon

Line 3 – Room states, followed by a colon

Line 4 – The last room that the user visited in the previous game

Line 5 – beenHere boolean, signifying if the user’s current room is the room they last entered in the previous game

Line 6 – Delimiter (---)

Key word Rooms, followed by a colon (:)

Multiple rooms until delimiter (===)

* Each room consists of:
  + Room name
  + beenHere boolean
  + Contents (items inside the room, separated by a comma)
  + Delimiter (---)