Object Oriented Optimists (OOO)
John
Richard Volynski
Robert
7/3/2020
CPSC 240
Prof. Davies

Zork++

Phase 1

Use Cases

Use Case 1

Name: Kill

Synopsis: In addition to armory and weapons, the .zork file will contain a special item called Snake in a specific room. When the user enters the room where the snake is, the system will print “There is a snake in this room.” The user will use the command “kill snake.” Then, the system will print a message “Kill snake with what?” If the user possesses a dagger or sword, they will type “kill snake with sword” or “kill snake with dagger.” The system will remove the snake from the room and print a message “After suffering life-threatening injuries, you manage to kill the snake” and user score will increase by 25 points. If the user does not possess a dagger or a sword, the snake will bite them and the system will print a message “The snake bit you!” and user score will decrease by half. If the user enters “health”, the system will print “You suffer intense pain while flesh comes out of your body.”

Actors: User

Precondition(s): The user has to enter the room where the snake is hiding and possess either a dagger or sword.

Sunny Day flow:

1. The user enters the room
2. Following the room description, there will be a message “There is a snake here”
3. The user will enter the command “kill snake” (case insensitive)
4. The system will print “Kill snake with what”
5. The user will type “Kill snake with dagger” or “Kill snake with sword” (case insensitive)
6. If user types “kill snake with dagger”, the system will print “The snake bites you in the ankle as you slash the snake’s tail off” and user score will decrease by 10 points. If user types “kill snake with dagger” again, the system will print “You attempt to stab the snake in the back as it bites your hand off” and user score will decrease by another 10 points. If the user types “kill snake will dagger” again, the system will print “After suffering multiple life-threatening injuries, you manage to kill the snake” and user score will increase by 25 points.
7. If the user types “kill snake with sword”, the system will print “You stab the snake in the back as it smacks its venomous tail in your face” and user score decreases by 10 points. If user types “kill

snake with sword" again, the system will print "After suffering life-threatening injuries, you manage to kill the snake" and user score will increase by 30 points.

Rainy Day #1

1. The user does not possess a dagger or a sword
2. The snake will bite the user and the system will print a message "The snake bit you!".
3. If the user enters "health", the system will print "In addition to the snake eating flesh in your left foot, you are suffering life-threatening injuries."
4. If the user moves into another room, the system will print a message "The snake bites you in the back as you attempt to flee" and user score decreases by 20 points
5. If the user enters "health", the system will print "You are crawling on one foot and one arm."

Use Case 2

Name: Swap

Synopsis: In addition to armory and weapons, there will be items spread throughout different rooms for the user to take. The swap command will replace items in the room the user is currently in with items the user possesses.

Actors: User

Precondition(s): The user needs to possess an item(s).

Sunny Day flow:

1. The user enters the room
2. Following the room description, there will be a message "There is a(n) [item] here"
3. The user will enter the command "swap" (case insensitive)
4. The system will print "Swap what? (usage: swap <item> with <item>)"
5. The user will enter "swap <item> with <item>"
6. The system will swap the user's item(s) with item
7. The user's inventory will be updated

Rainy Day #1

1. The user doesn't have any items

2. The system will print “You don’t have a(n) <item> to swap” or “You don’t have any items to swap”

UML Diagrams

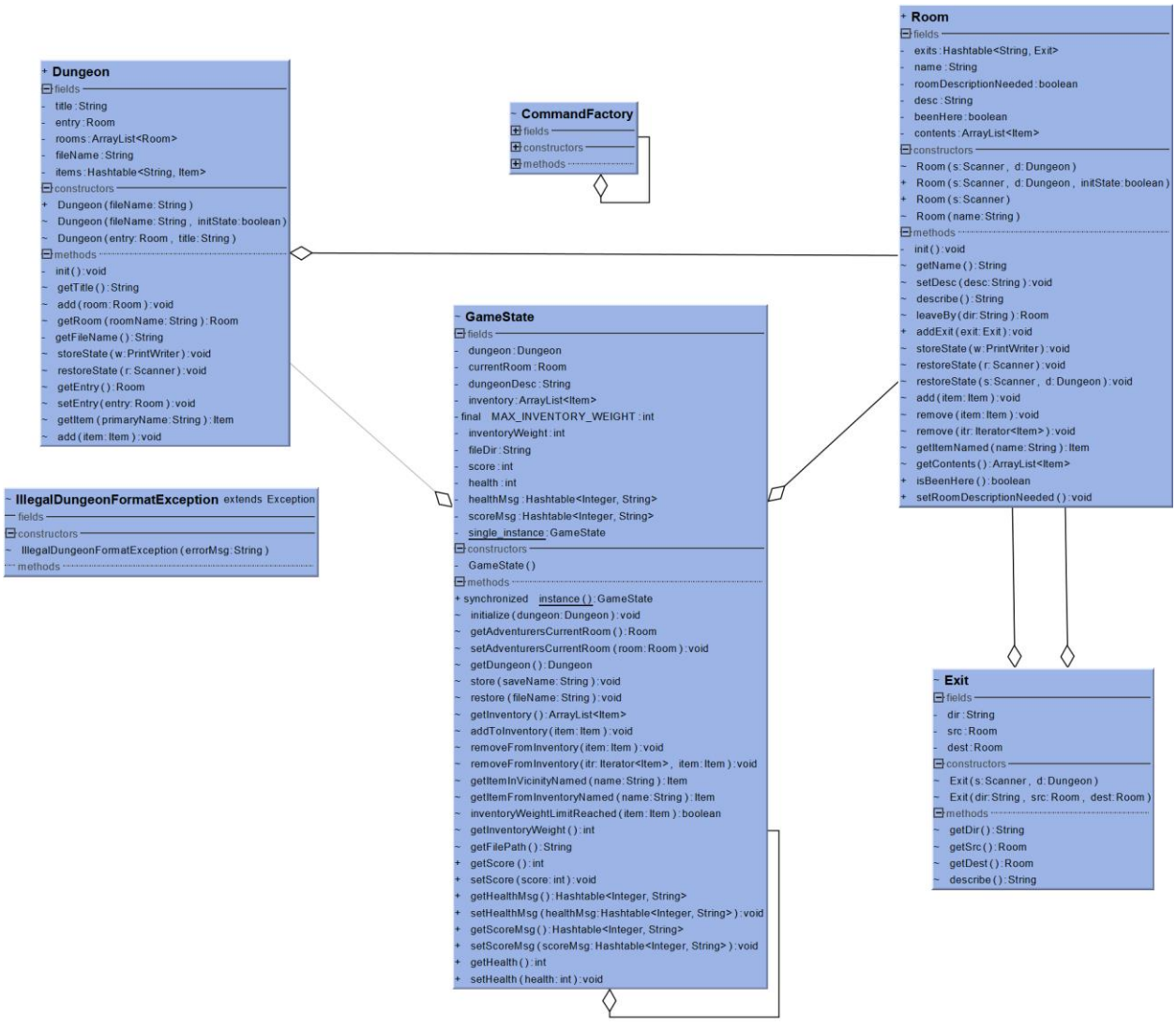

Figure 1: Zork++ Main classes diagram

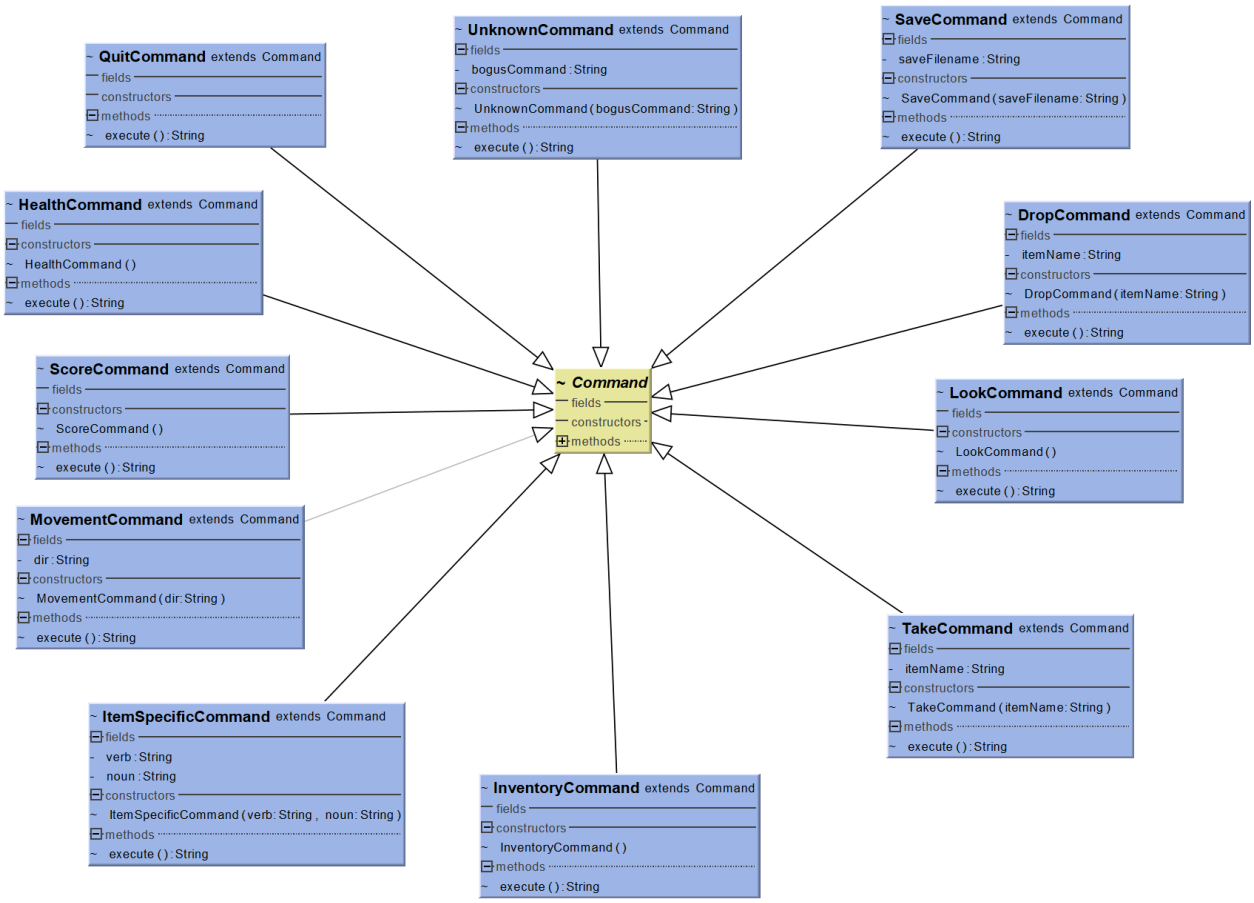

Figure 2: Zork++ Command classes diagram

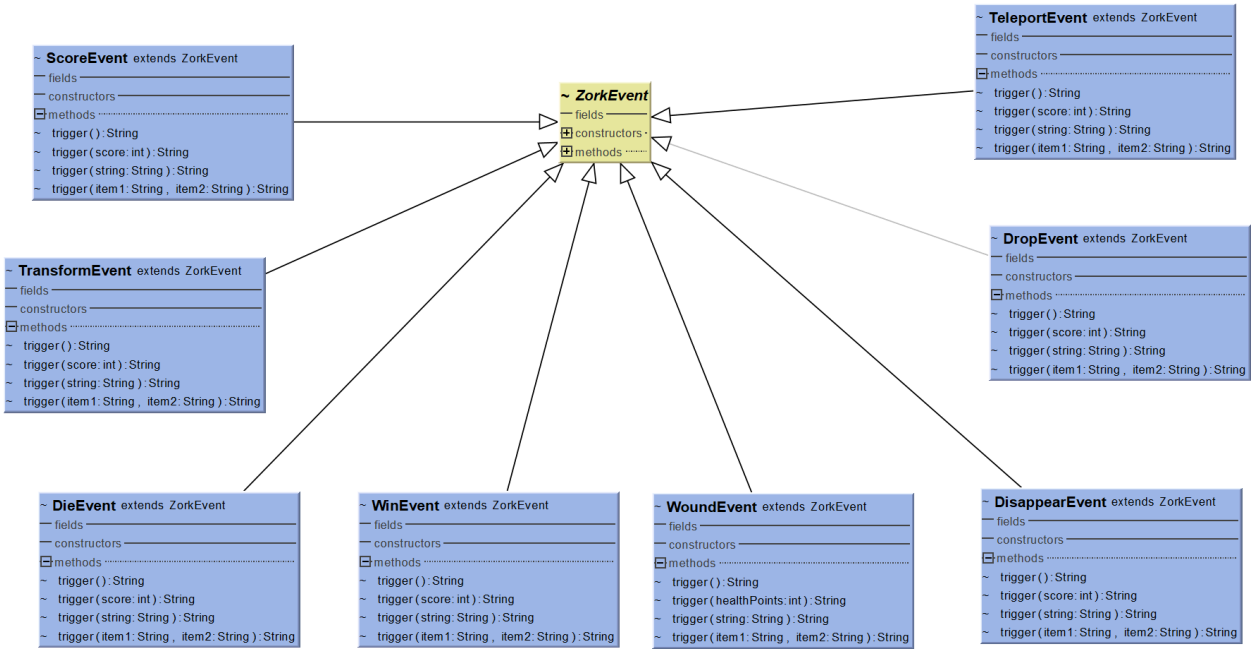

Figure 3: Zork++ ZorkEvent classes diagram

Sequence Diagrams

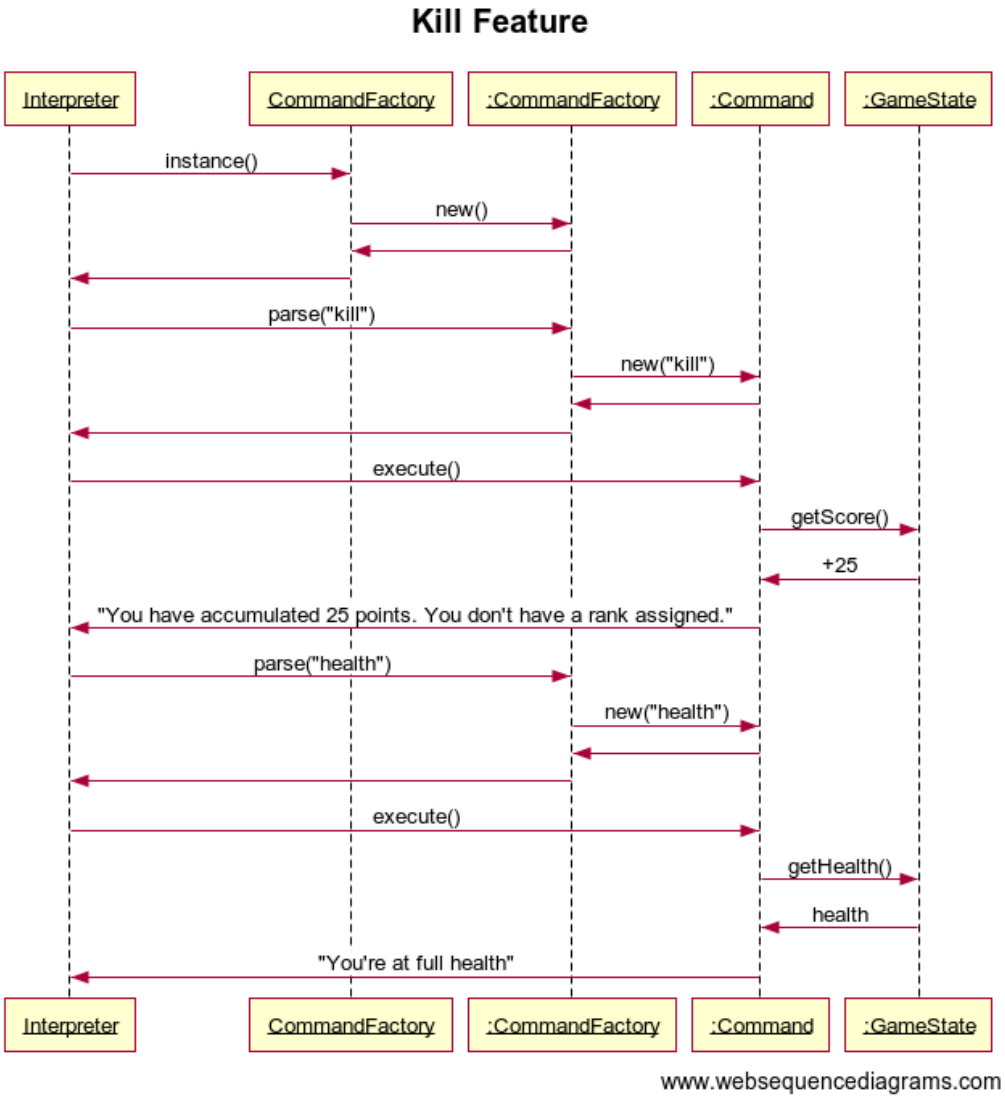

Figure 4: Supplemental Feature Kill Sequence Diagram

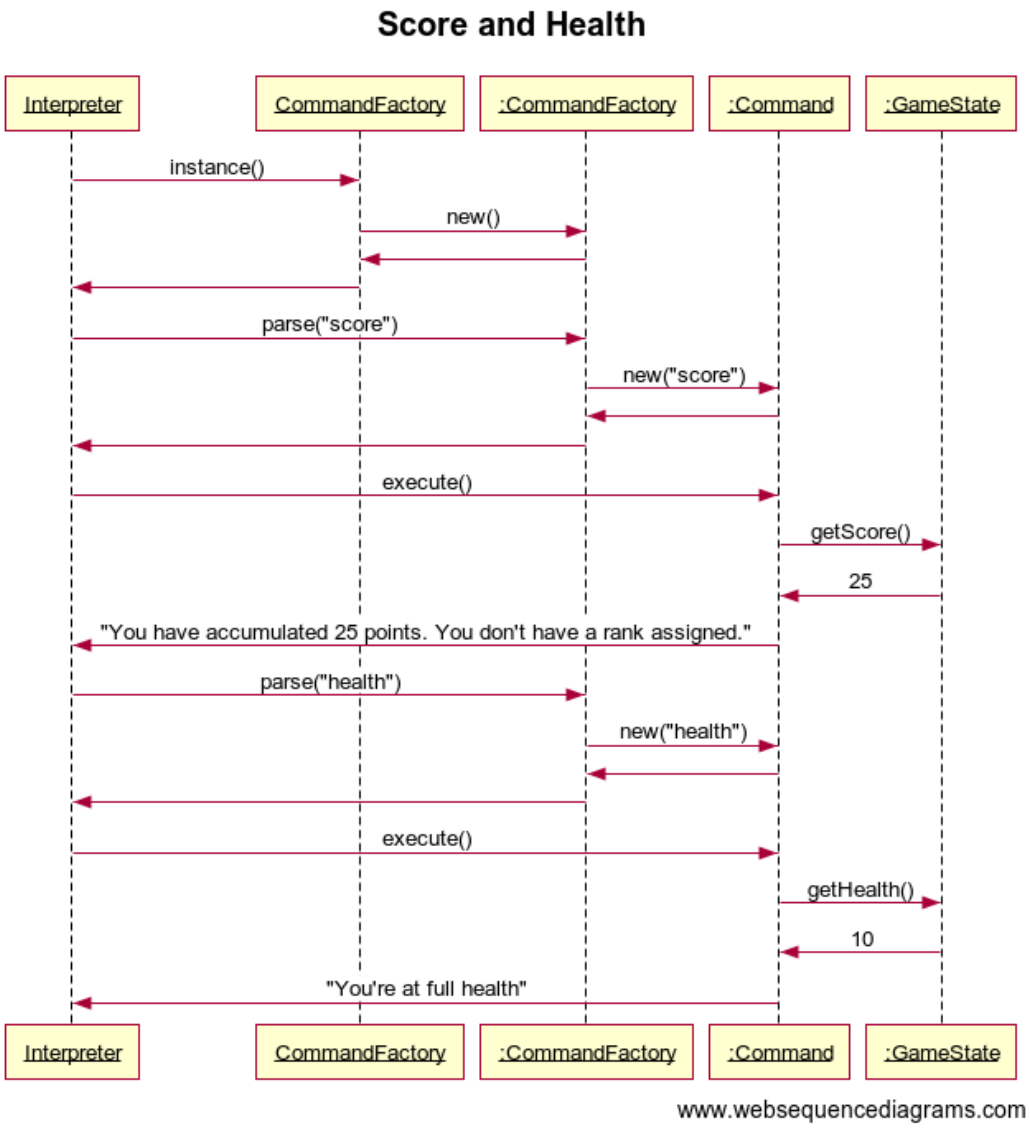

Figure 5 Supplemental Feature Score and Health Sequence Diagram

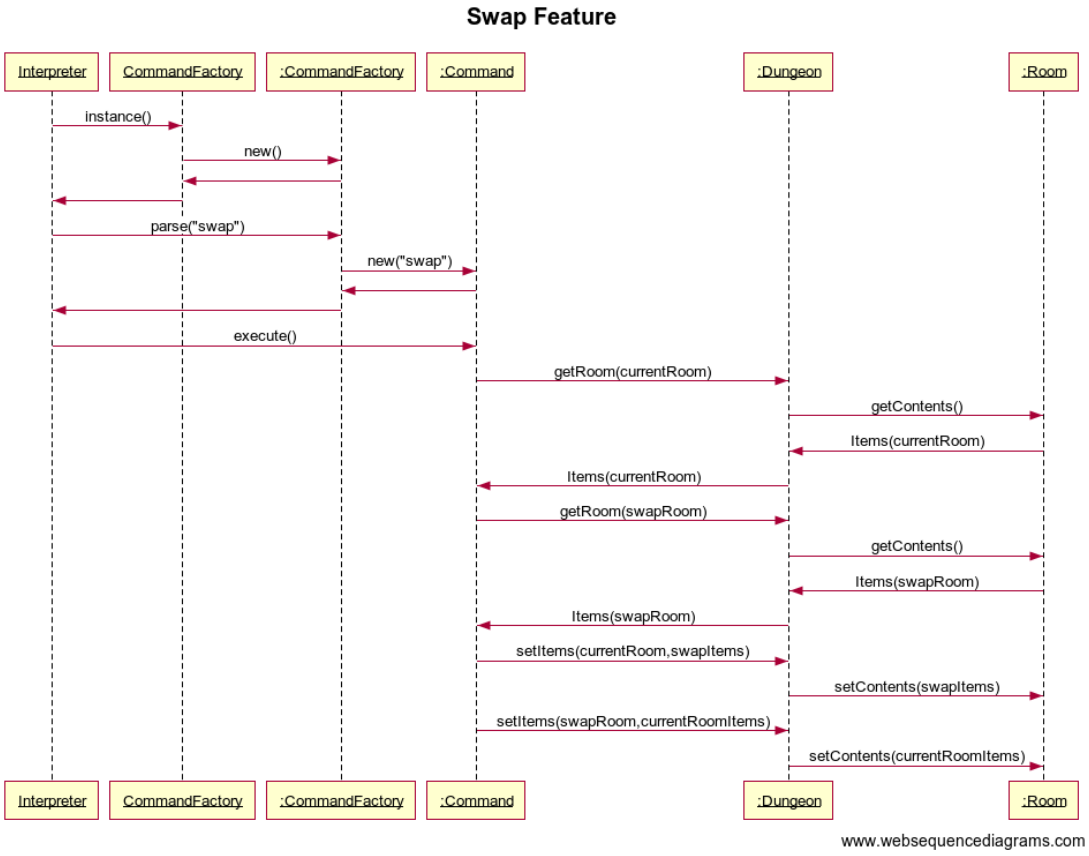

Figure 6: Supplemental Feature Swap Sequence Diagram

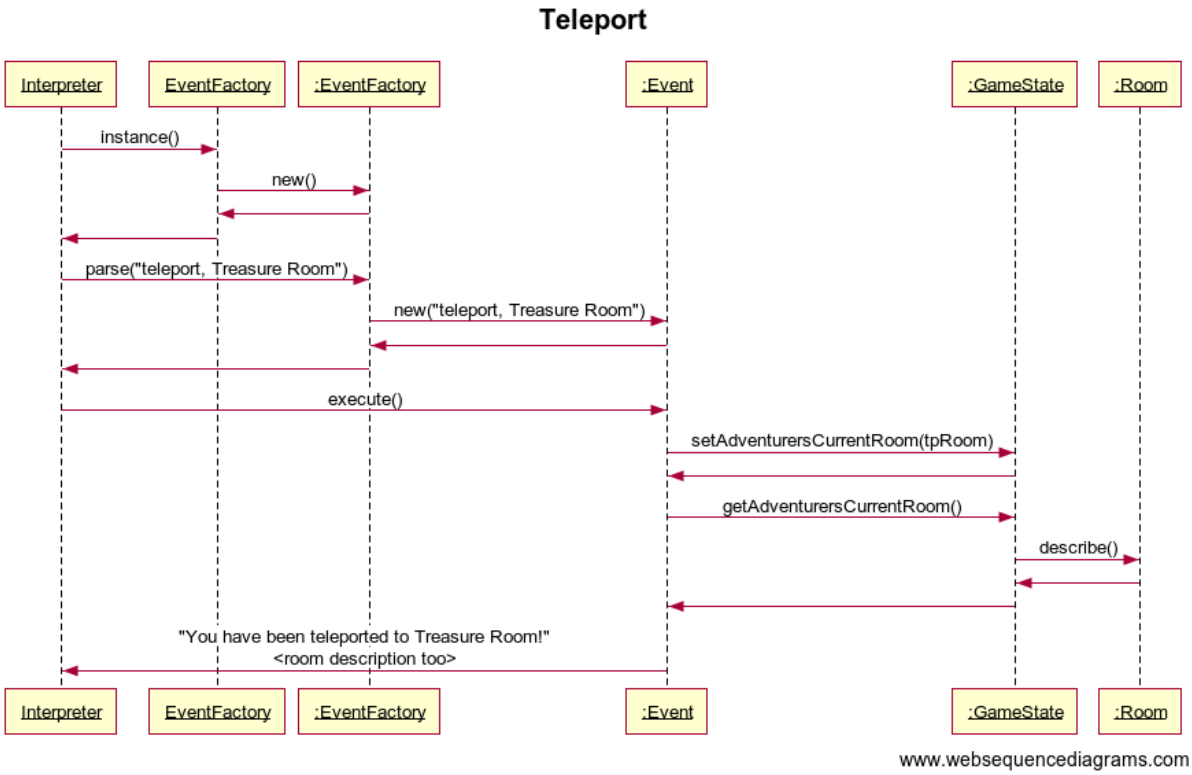

Figure 7: Supplemental Feature Teleport Sequence Diagram

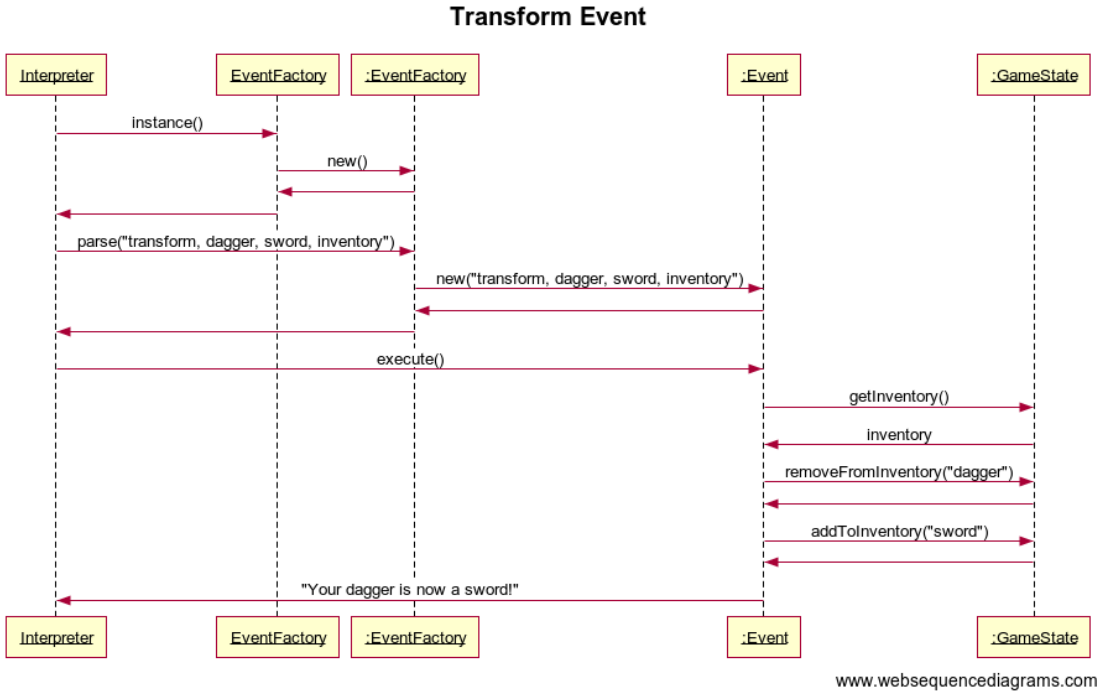

Figure 8: Supplemental Feature Transform Sequence Diagram

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 – Key word Room states, followed by a colon

Multiple rooms until delimiter (===)

- Each room consists of:
  - Room name
  - beenHere boolean, signifying if the user's current room is the room they last entered in the previous game

- Room contents (Key word Contents: items inside that room, separated by a comma), if applicable
- Delimiter (---)

Key word Adventurer, followed by a colon

Key word Current room, followed by a colon and the name of the room the user last entered in the previous game

Key word Inventory, followed by a colon and the items the user possesses, separated by a comma

Key word Score, followed by colon and the user's score for the previous game

Key word Health, followed by colon and user's health in the previous game (measured in hp)