Interaction diagrams for System Finding Trump

Assignment in the course PA1435 Object Oriented Design

2017-04-25

Author Name	Social Security#	Thinking	Writing
Emil Wahl	880729-0351	17%	17%
Fredrik Carlsson	961206-0070	17%	17%
Daniel Cheh	941231-1772	17%	17%
Aksel Grahn	940311-0159	17%	17%
Sebastian Vusak	960406-1110	17%	17%
Gustav Björk	960223-4131	17%	17%

System Description

The system is based on the classic game Nethack. Nethack is a text-based and two-dimensional game. It is a rogue dungeon crawler with a random generated map. In this dungeon you will be able to find treasure and fight monster.

The main difference between our system and Nethack is that our system is connected to Twitter and all the levels, items, and monsters are generated from a seed in the form of a Twitter feed.

Prioritised List of Use Cases

Name	Story Points
Generate level	32p
Move around	6p
Pick up item	3p
Winning conditions	1p
Show character Information	2p
Fight Monster	10p
Losing conditions	1p
Equip item	1p
Open chest	1p
Open door	1p
Use key	1p
Character creation	5p
Merchant interaction	1p
Buy item	1p
Sell item	1p
New game	1p
Game quit	1p
Save game	1p

Load game	1p
Drop item	3p
Join game	50p
Chatting between players	10p

System events

1. System events for use case Generate level
   generate_Level()

2. System events for use case Move around
   move()

3. System events for use case Pick up item
   add_Item()

4. System events for use case Winning conditions
   trump_Check()

5. System events for use case Show character information
   show_Info()

6. System events for use case Fight monster
   combat_Action()
   damage_Check()

7. System events for use case Losing conditions
   player_Death()

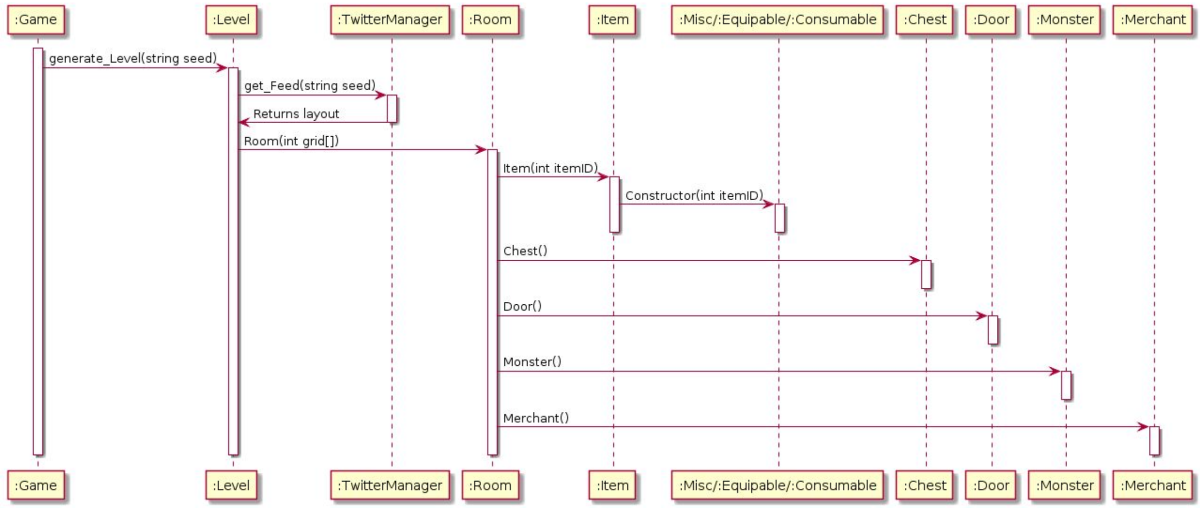
8. System events for use case Open chest
   open_Chest()

9. System events for use case Open door
   open_Door()

Interaction Diagrams

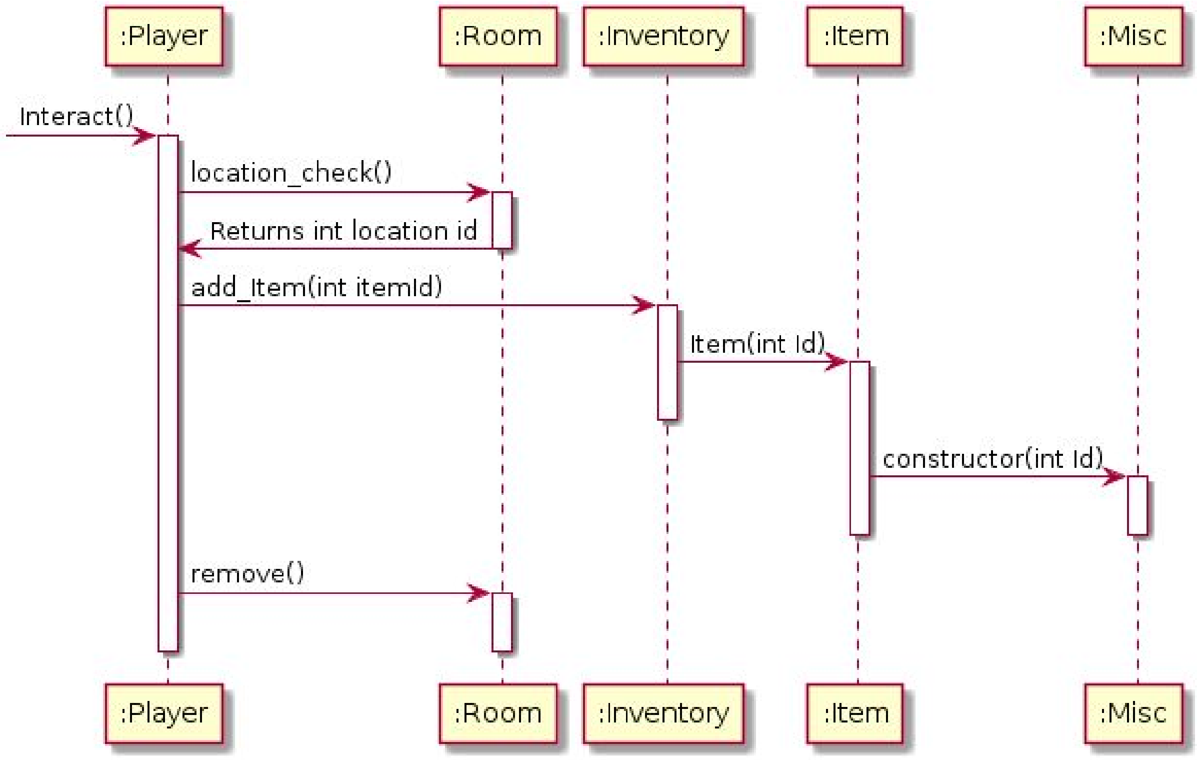
Interaction diagrams for Use Case Generate Level:

System event generate_Level()



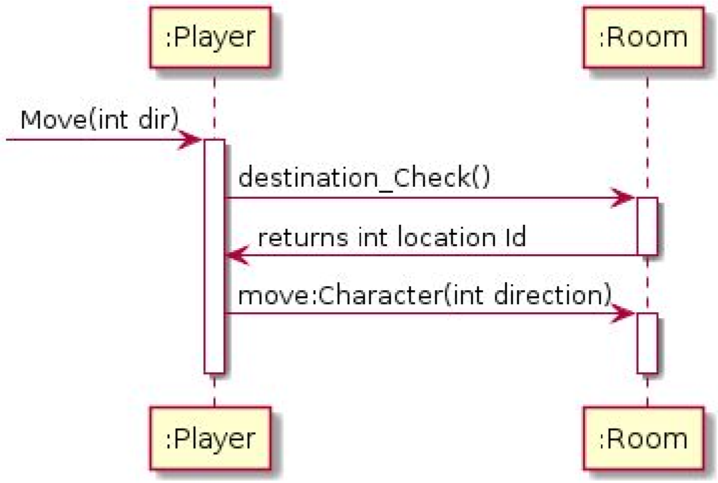
Interaction diagrams for Use Case Pick Up Item:

System event add_Item()



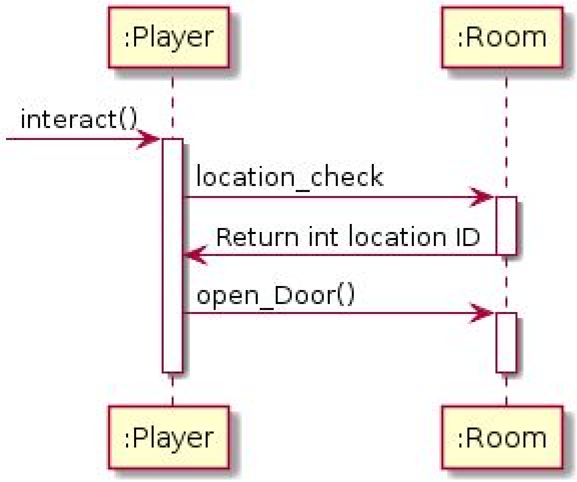
Interaction diagrams for Use Case Move Around

System event move()



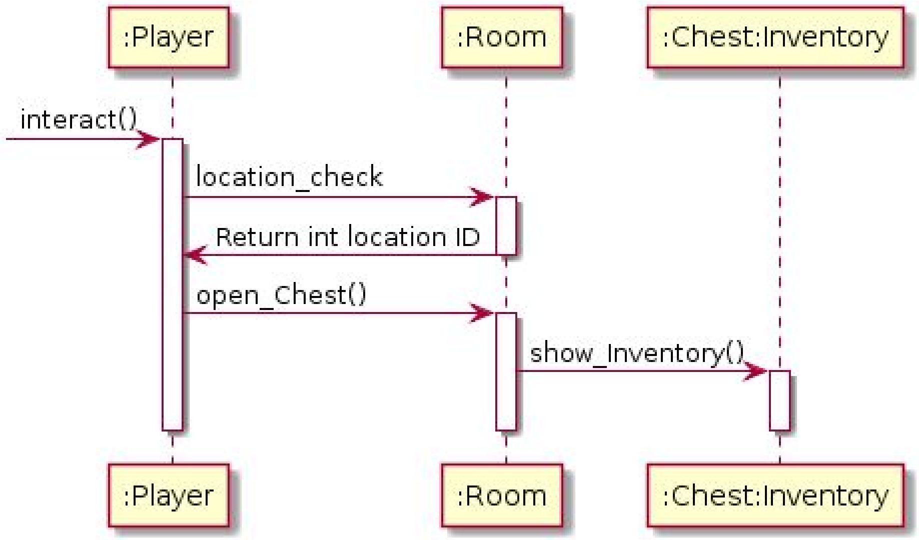
Interaction diagrams for Use Case Open Door

System event open_Door()



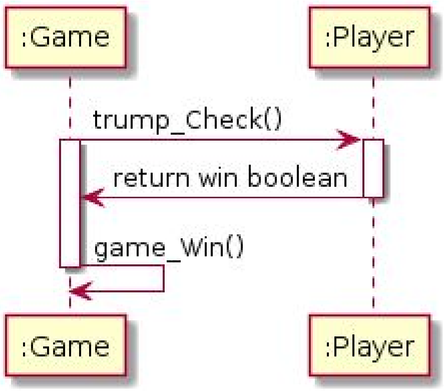
Interaction diagrams for Use Case Open Chest

System event open_Chest()



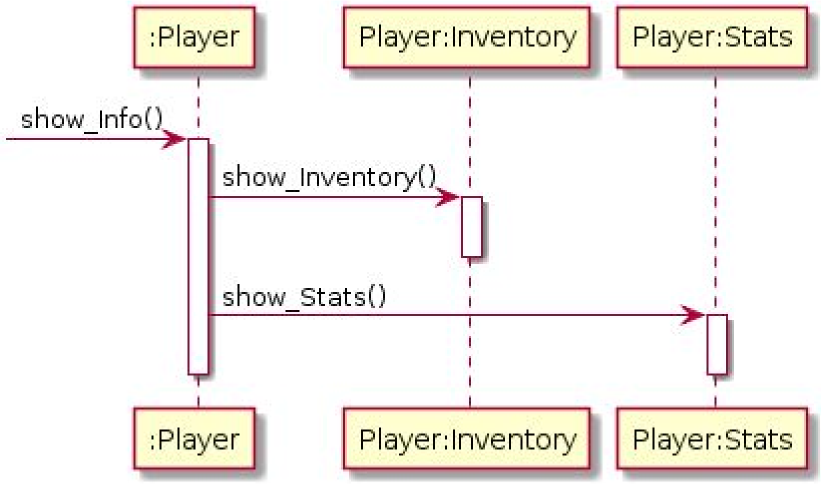
Interaction diagrams for Use Case Winning Condition

System event trump_Check()



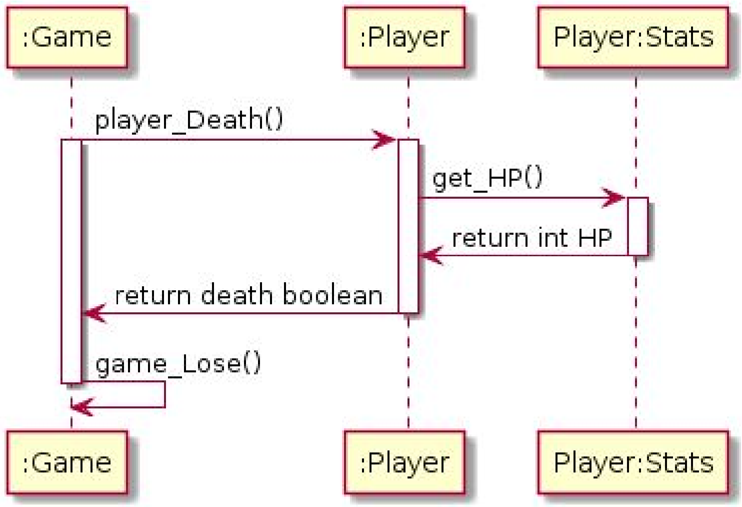
Interaction diagrams for Use Case Show Character information

System event show_Info()



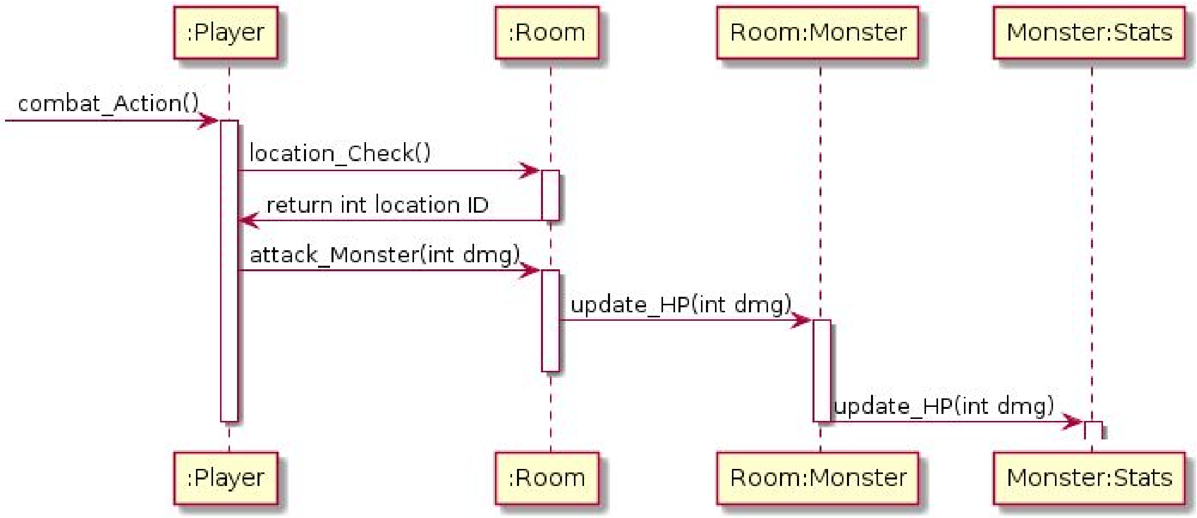
Interaction diagrams for Use Case Losing Condition

System event player_Death()

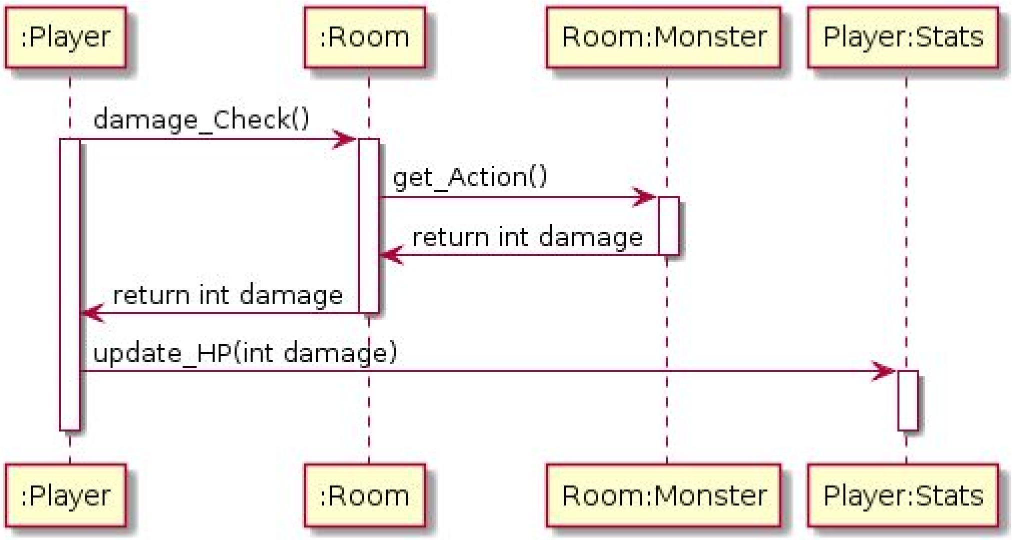


Interaction diagrams for Use Case Fighting Monsters

System Event combat_Action()



System event damage_Check()


:Player
:Room
Room:Monster
Player:Stats
damage_Check()
get_Action()
return int damage
return int damage
update_HP(int damage)
:Player
:Room
Room:Monster
Player:Stats