

Library

Software Engineering Homework

Kopach Artem

Neptun code: JQBOLI

1 DESCRIPTION OF THE TASK

LIBRARY

In an underground library, monks are looking for the Holy Grail. The rooms of the library are connected by doors. Monks can go from one room to a neighboring room through a door. A room may have any number of neighboring rooms. Some doors are one-way only.

Rooms may contain special relics. One of these relics is the Holy Grail itself. The monks can pick up the relics, but they can only carry 5 relics at a time. The monks can also put down relics. The monks can also write on the walls and they can read these writings.

Daemons are trying to stop the monks to be successful. If a daemon meets a monk (they are in the same room), the daemon takes the soul of the monk and the monk dies. However, there are relics which can protect against daemons. The saint cloak protects the monk three times, and after that, it loses its protecting ability. The saint ring protects only for a while, and after this time expires, it loses its protecting ability. The saint smoker also works only for a while, and it destroys all the daemons within the same room.

Each room has a capacity, which can be any positive integer number. No more monks can enter a room if the room's capacity is reached.

There are multiple kinds of rooms. When a daemon enters the "daemon mover" room, the room immediately transfers the daemon to a neighboring room. This room has no effect on monks. The "poisonous room" contains a special poison which makes the monks entering the room unconscious for a while. The doors of the "accursed room" sometimes vanish, but after a while, the doors reappear. The players control the monks and they win if they pick up the Holy Grail.

Ambiguous requirements or any requirement not precisely described above can be freely customized as you like. Usually, it is best to choose the easier solution. Numbers cannot be limited from above (e.g. when "multiple" or "at least" or "any number" appears in the requirement).

2 FUNCTIONAL REQUIREMENTS

2.1 PRIMARY REQUIREMENTS

Identifier	Description	Use case
R1	Player control some monk and move them	Move monks, view space
R2	Rooms connected by doors. A room may have any number of neighboring rooms. Some doors are one-way only.	view space, move monks, daemons action
R3	Rooms may contain special relics	Pick up relics, view space
R4	One of these relics is the Holy Grail	Pick up relics
R5	The monks can pick up the relics, but they can only carry 5 relics at a time.	Pick up relics, move monks
R6	The monks can also put down relics	Put down relics
R7	The monks can write on the walls messages	Write message
R8	The monks can also can read these messges.	Read message
R9	Daemons are trying to stop the monks. Attack them	daemons action
R10	If daemon and monk are in the same room, the daemon takes the soul of the monk.	daemons action
R11	Monks haave a relics, which can protect against daemons	move monks, pick up relics
R12	One of them (relics) the saint cloak protects the monk three times, and after that, it loses its protecting ability	move monks, pick up relics
R13	Another, the saint ring protects only for a while, and after this time expires, it loses its protecting ability.	move monks, pick up relics
R14	the saint smoker also works only for a while, and it destroys all the daemons within the same room.	move monks, pick up relics
R15	Every room has a capacity	move monks, daemons action, view space
R16	When daemons attack monk. The monk starts defending and uses all relics by order, which he/she has.	
R17	No more monks can enter a room if the room's capacity is reached.	move monks, view space
R18	There are multiple kinds of rooms	view space, move monks, daemons action
R19	The "daemon mover" room, the room immediately transfers the daemon to a neighboring room. This room has no effect on monks.	move monks, daemons action, view space
R20	The "poisonous room" contains a special poison which makes the monks entering the room unconscious for a while.	move monks, daemons action, view space

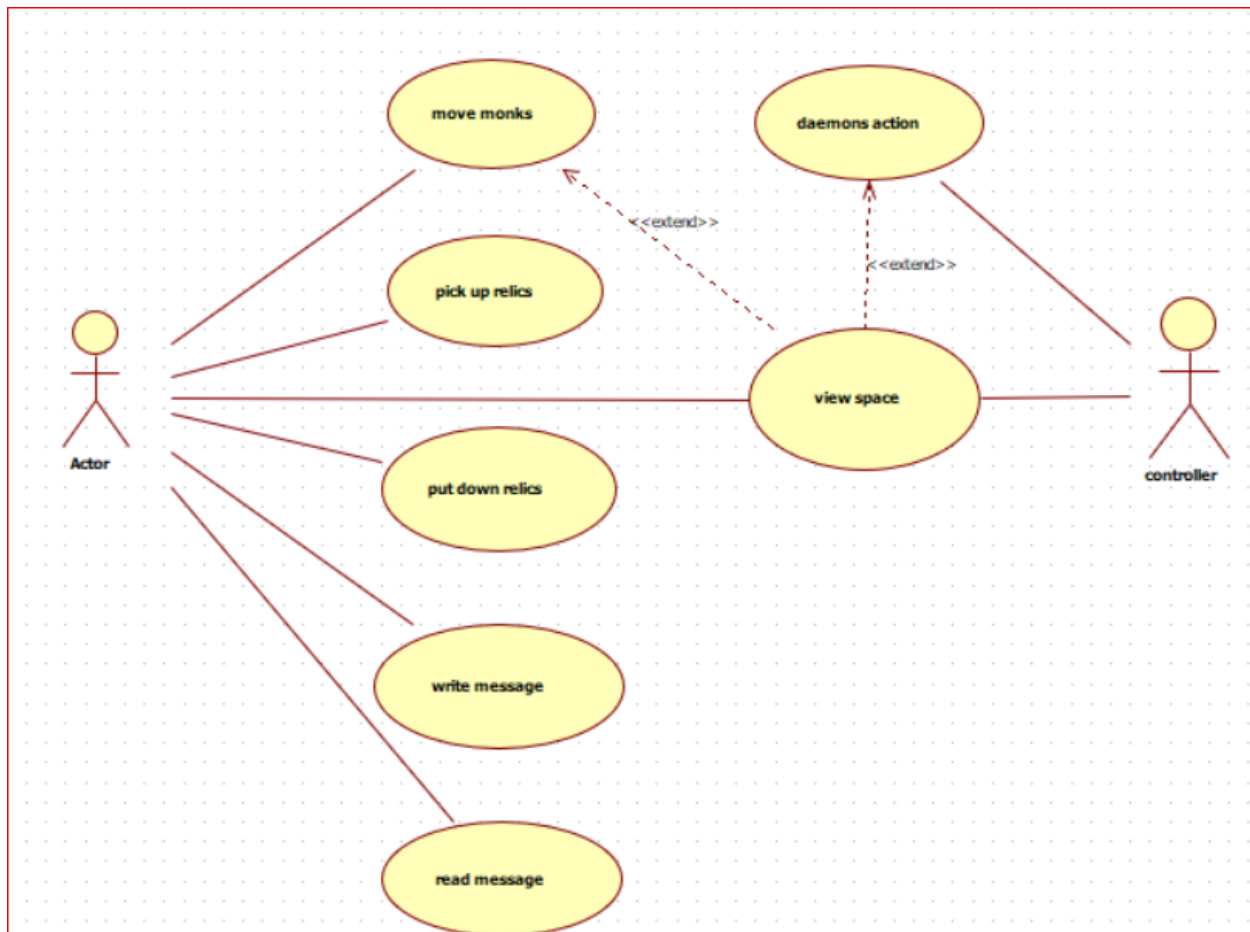
R21	The doors of the “accursed room” sometimes vanish, but after a while, the doors reappear.	move monks, daemons action, view space
R22	The players control the monks and they win if they pick up the Holy Grail.	move monks, pick up relics
R23	If the monk successful defend, then or nothing happened, or if relic could kill daemon, he will die.	move monks, daemons action

2.2 ADDITIONAL REQUIREMENTS

Identifier	Description	Use case
R24	Move single monk to the select room. Controller will check capacity, active door or no	move monks, view space.
R25	Daemons move to the room like monks.	daemons action
R26	Relics have 4 different types: the saint cloak, the saint ring, the saint smoker and Holy Grail	pick up relics
R27	Capacity can be any positive integer number	move monks, daemons action, view space
R28	Controller won't try to move daemons to room, which has maximum capacity	daemons action
R29	Room has 3 type daemon mover room, “poisonous room”, “accursed room”	view space, move monks, daemons action
R30	Before all attacks, when somebody switch place room do them action to daemons or monks	move monks, daemons action

3 USE CASES

3.1 USE CASE DIAGRAM



3.2 USE CASE DESCRIPTION

Title	move monks
Description	The player move monk to neighbor room
Actors	Actor
Main success scenario	The palyer move monk to the target room, which is connected with current.
Alternate scenario	The palyer does not move, if the door is not active
Alternate scenario	
Alternate scenario	When monk moves to target room, room start do kinf of action(depend by kind of rooms)
Alternate scenario	If the capacity in the target room is maximum (=5), then monk stay at the current room and player may select another room

Title	pick up relics
Description	The palyer pick up relic from room
Actors	Actor
Main success scenario	The player try to pick up some relic, relic remove from room and add to monak
Alternate scenario	If the monak has 5 relics, he/she can not collect more
Alternate scenario	If palyer(monk) collect Holy Grail in current room, then game is end
Alternate scenario	-
Comment	Monak start to collect relic, when he is free. If monak defend he can not collect them

Title	put down relics
Description	The player put down some relic
Actors	Actor
Main success scenario	The player remove relic, and this relic add to the room
Alternate scenario	If monak does not have any relic, player won't drop something
Alternate scenario	-
Comment	monk can drop relics all time, when he/she would like to do it

Title	write message
Description	Monak write message
Actors	Actor
Main success scenario	The player by monak write message to the wall and this message had been added to the wall in current room
Alternate scenario	-

Title	read message
Description	Monak read message
Actors	Actor
Main success scenario	The palyer by monak read message from the wall and get infromation from them
Alternate scenario	-
Comment	All rooms have any messages

Title	view space
Description	Player can view space and controller can desribe space
Actors	Controller, Actor
Main success scenario	The player get inform about space, before move monak. Controller also check space, before do „daemons action“

Alternate scenario	-
--------------------	---

Comment for Use Case Diagramm: I add <<extend>> before move monks and move daemons, because, before this move we should know information about space

4 STRUCTURAL DESCRIPTION

4.1 DESCRIPTION OF THE CLASSES

Comment for Class Diagram before: NB that first of all all objects is alive for all time in the game, and they just swap states in the period of game (not die for real). Also, this is a little bit simpler model.

For all compositions, generalizations and associations I didn't wrote elements, which is get by the relation between two objects, because it is clearly.

4.1.1 Room

Responsibilities

This is parent class for all kind of rooms, also contain all attributes and methods, which must be in the subclasses.

Attributes

+wallMessage: List <String>	This will save all message in the room (<<read only>>)
+capacity: unsigned int	This will store capacity of room(comment: this is positive integer, that is why hear unsigned int). Can be any positive number

Methods

+monksList() : List<Monks>	Will show how many monks in this room
+daemonList(): List<Daemons>	Will show how many daemons in this room
+relicsList(): List<Relics>	Will show which relics in the room now
+messageWrite(m: String)	Will add message to the walls on the room
+isRoom(): boolean	Will show how many space in the room (Comment: valid by capacity condition or no)
+roomAction(): boolean	This function describe room Action for current room (interface)
+getDoors(): List<Doors>	This is will get the all ACTIVE doors in current room
+addRelic(r: Relics)	This is will put relics to the room, when monk will drop this
+addMessage(m: String)	This is will put message to the wall, when monk will write this
+readRoomMessages(): List<Srteing>	This is will read all messages from room

+removeRelict(r: Relics)	This is will remove relic from current room, when we pick up this

4.1.2 Poisonous room

Responsibilities

Subclass of room, with special realization roomAction()

Attributes

Come from generalization (inharitance)

Methods

+roomAction()	For this type of room the function will special poison which makes the monks entering the room unconscious for a while

4.1.3 Daemon mover

Responsibilities

Subclass of room, with special realization roomAction()

Attributes

Subclass of room, with special realization roomAction()**Methods**

+roomAction()	This is type of room will immediately transfers the daemon to a neighboring room. This room has no effect on monks

4.1.4 Accursed room

Responsibilities

Subclass of room, with special realization roomAction()

Attributes

Come from generalization (inharitance)

Methods

+roomAction()	This is type of room will sometimes vanish, but after a while, the doors reappear

4.1.5 Door

Responsibilities

The class which describe doors in the game. And get inform about active door or no

Attributes

-	
-	

Methods

+isActive(): boolean	Will get infrom is door active or stack

4.1.6 Monks

Responsibilities

Monk class, will realise all olayer features in the game and describe this.

Attributes

-relicsList: List<Relics>	Will collect which relics monk has
- unconscious: boolean–Step??	Will do unconscious monk for some time
-isLive: boolean	Live or not monk (Comment: after daemon attack)
+unconsionsTimeOut	Will store the time for what monk is unconsions

Methods

+monkDefend(): boolean	Monk will defend by relics (Comment: result 1:monk kill daemon, 0: died/has been defeet)
+monkMove(d: Door)	Monk will move to select door
+pickUpRelic()	Monk will pick up relic in the room (current room??)
+dropRelics()	Will drop relics from monk to room
+WriteMessage(m: String)	Monk will write some message to room wall

+readMessage(): List<Message>	Monk will read one or more messages from current room
+getDoors(r: Room): List<Doors>	Get doors in the current room
+removeRelic(r: Relics)	Remove choose relic from monak
+Step()	Realise step, because we need to know this in unconsions state
+kill()	Kill self, when monak lose defending
+changeRoom()	Will change room to the target room, whem player try to do this

4.1.7 Relics

Responsibilities

Parent class for all relics, realise all methods what they need

Attributes

+lastDefend: boolean	Status of the last defend

Methods

+relicDefend(): boolean	Relic defend until monak has relics

4.1.8 Saint clock

Responsibilities

Subclass, with special realisation of relicDefend()

Attributes

Methods

+relicDefend(): boolean	cloak protects the monk three times, and after that, it loses its protecting ability.

4.1.9 Saint ring

Responsibilities

Subclass, with special realisation of relicDefend()

Attributes

Methods

+relicDefend()	protects only for a while, and after this time expires, it loses its protecting ability.

4.1.10 Saint smoker

Responsibilities

Subclass, with special realisation of relicDefend()

Attributes

Methods

+relicDefend(): boolean	works only for a while, and it destroys all the daemons within the same room.

4.1.11 Holy Grail

Responsibilities

Class of relic, which can win a game

Attributes

-

Methods

-

4.1.12 Daemon

Responsibilities

Attributes

-isLAlive: boolean	Store status about daemon(Comment: if monak defended and kill then isAlive == 1, and daemon die)

Methods

+daemonMove(d: Door)	Move daemon to another room (target room, controller control this)
+daemonAction()	Daemon action ex., attack monak (automatically by controller)
+attack(m: Monks)	Attack monks in current room
+changeRoom()	Change room function for daemons
+selfKill()	When monak successfully defend monak kill daemon and daemon selfKill

4.1.13 World

Responsibilities

Just a container of all game with composition with all „main” classes

4.1.14 Land

Responsibilities

This is class which activate end of the game, when somebody get Holy Grail

Attributes

Methods

+winGame()	Will activate end of the game, when somebody collect Holy Grail
------------	-----------------------------------------------------------------

4.1.15 Timer

Responsibilities

Represents time in the game, makes the Steppables to step.

Attributes

-steppable: Steppable[0..*]	The steppables

Methods

+Tick()	Make all steppable step

4.1.16 Steppable

Responsibilities

An interface that represents anything which can step in time.

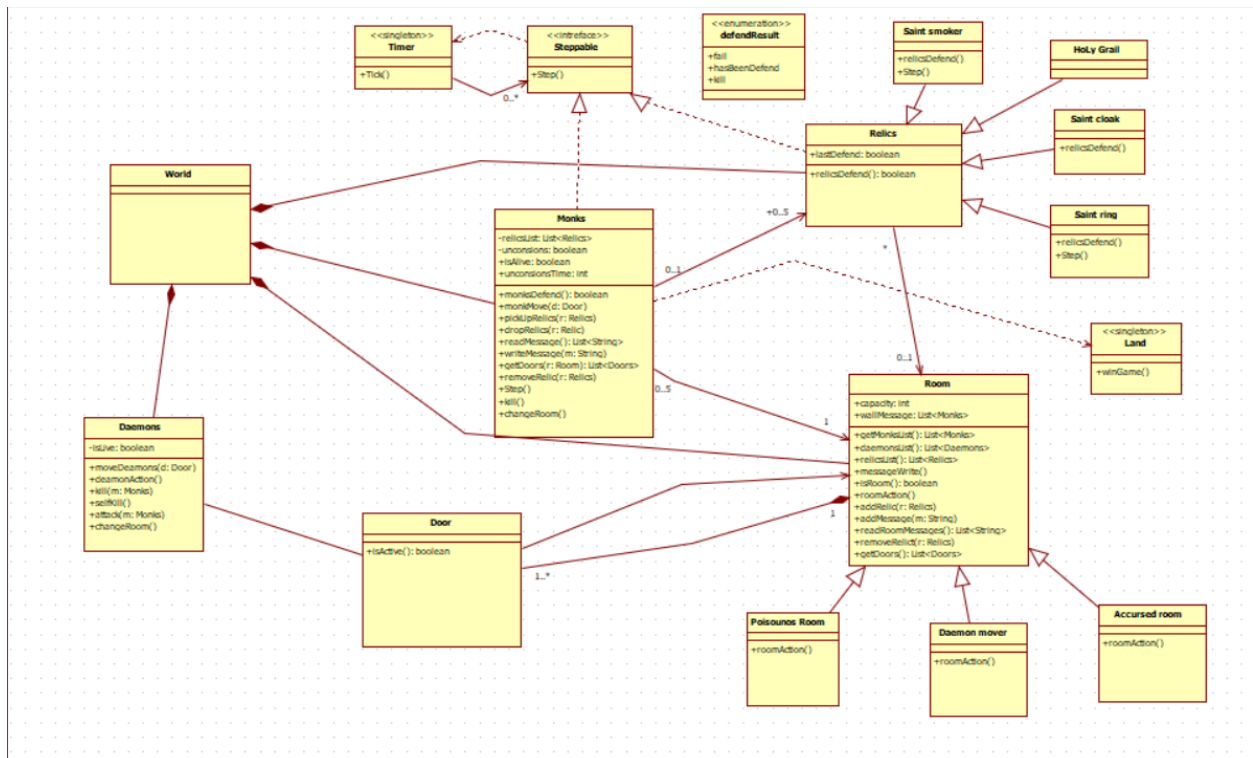
Attributes

-

Methods

+Step()	The action to perform a time step

4.2 CLASS DIAGRAM



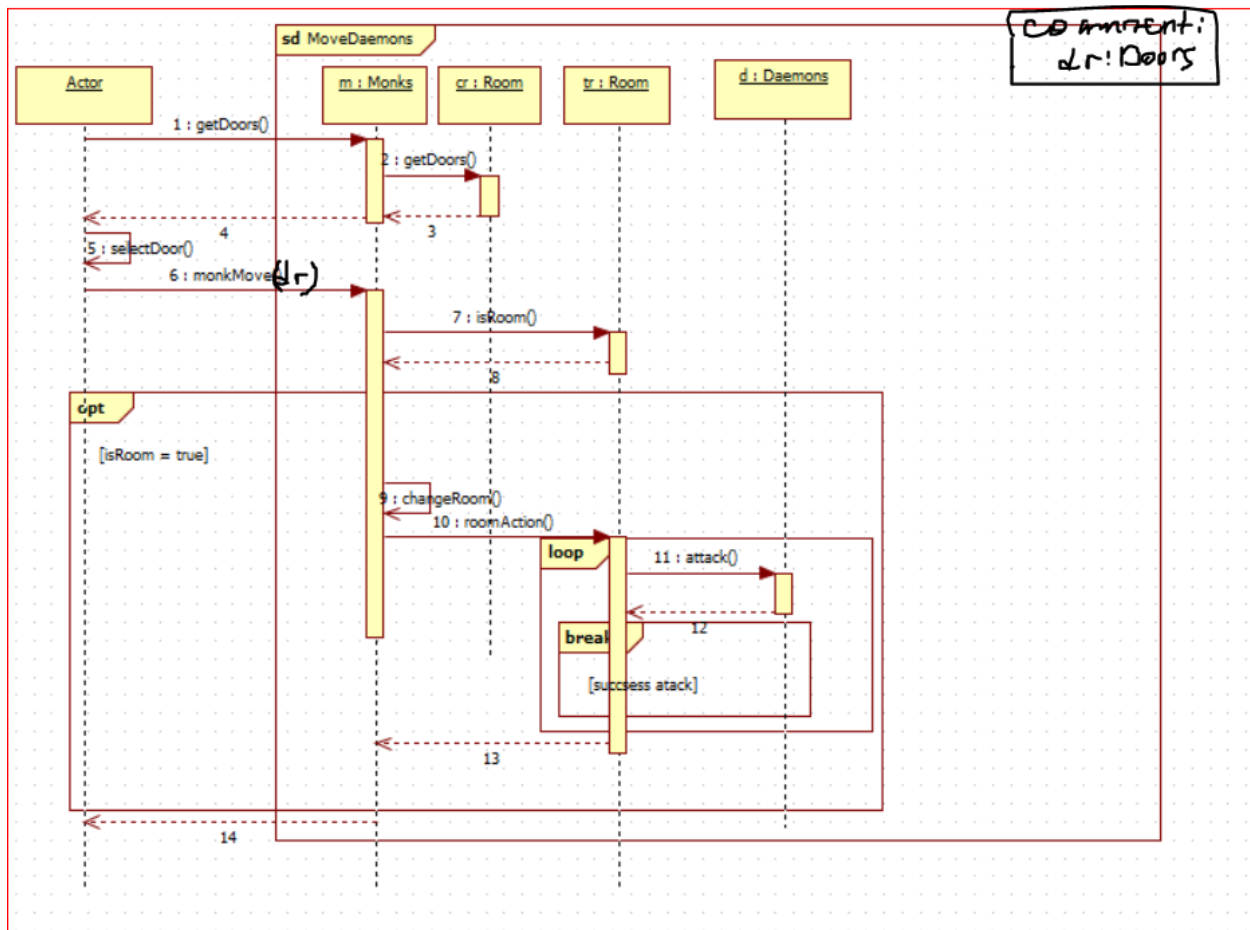
5 BEHAVIORAL DESCRIPTION

5.1 SEQUENCE DIAGRAMS

Comment: I have some problem with argument in the WhiteStarUML, that is why I wrote by hands argument. I did like in the tutorial and activate target class operations, but it does not show arguments

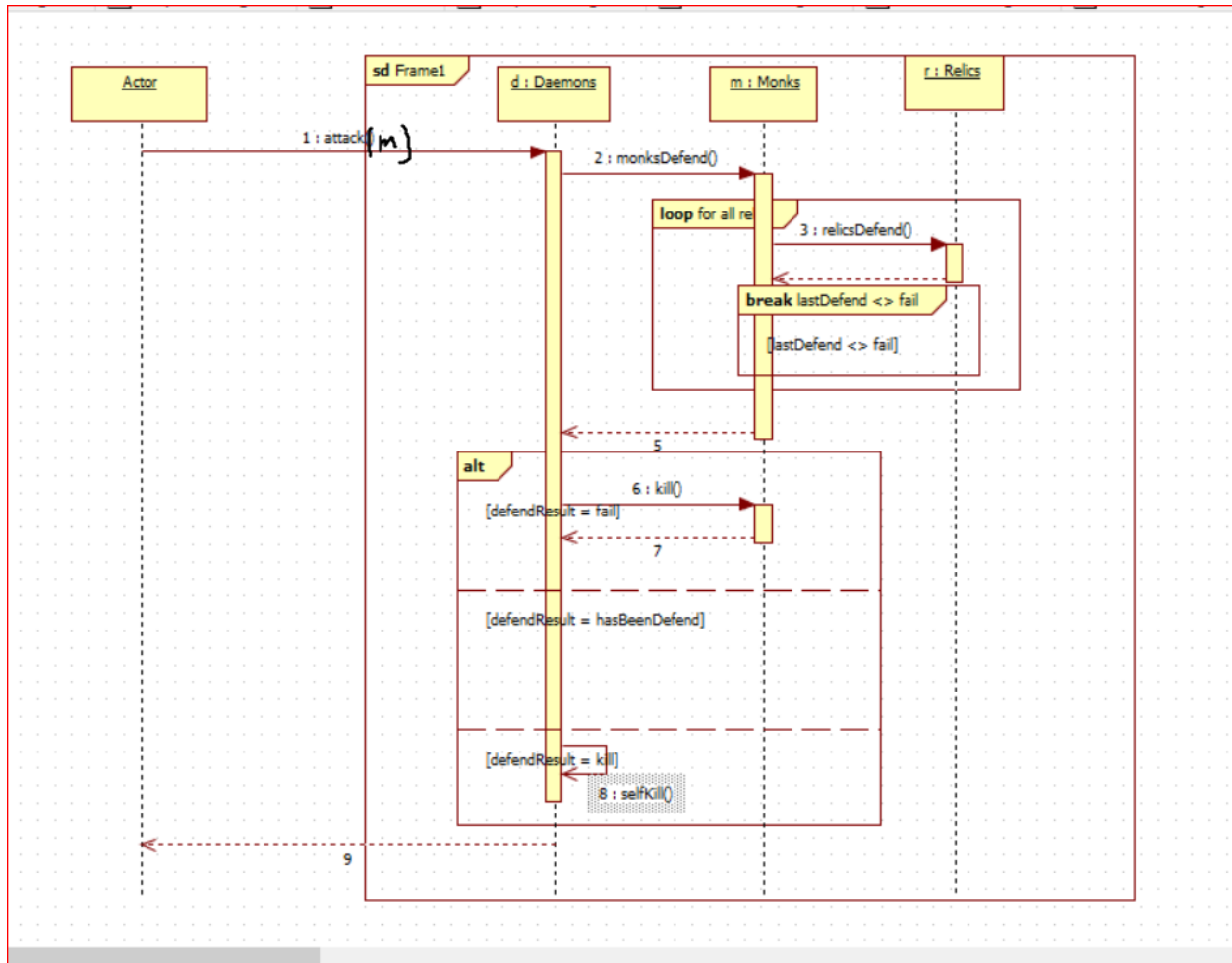
Also, I would like to detect a lot of problems in the program working. Sometime it does not show properties, lag and don not move some fragments!

5.1.1 Move monks



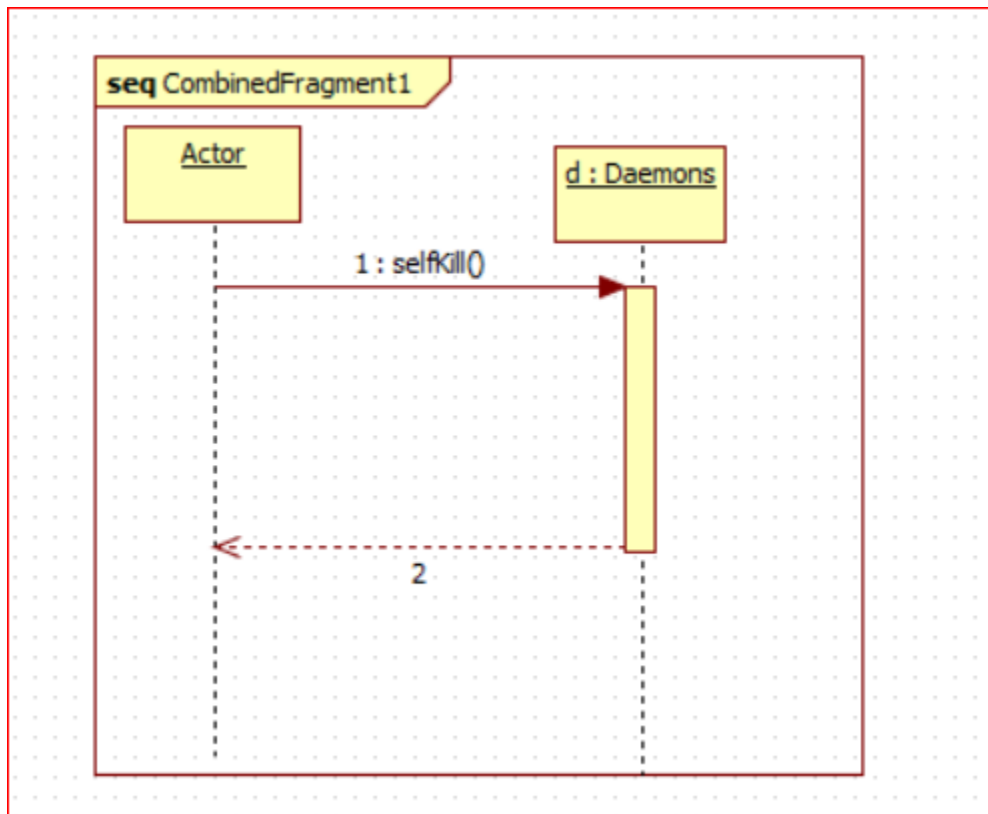
5.1.2 Attack

I do personal attack independently, for more readable

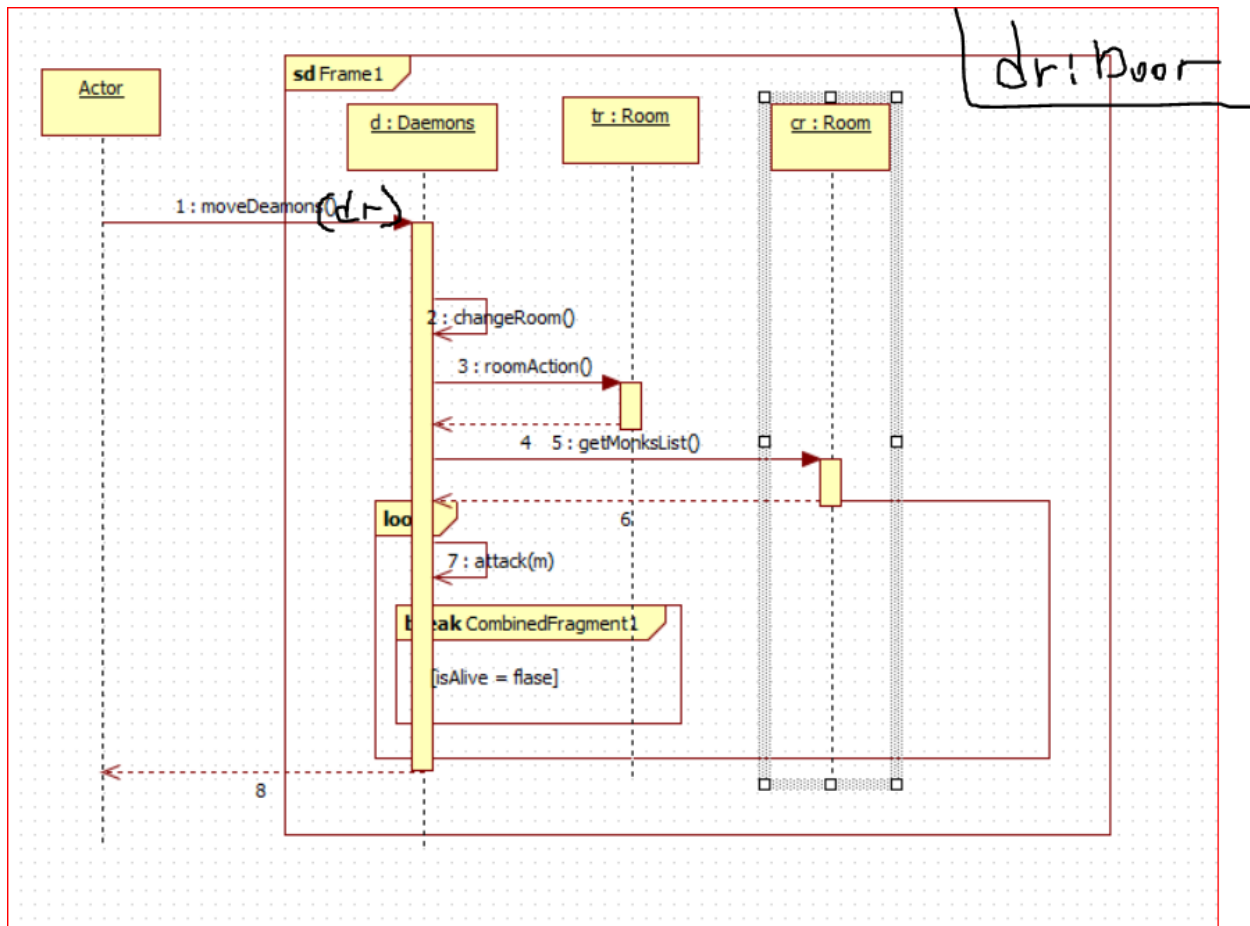


m is argument Monks, and selfKill I will show after is function which do isAlive = fail for daemon

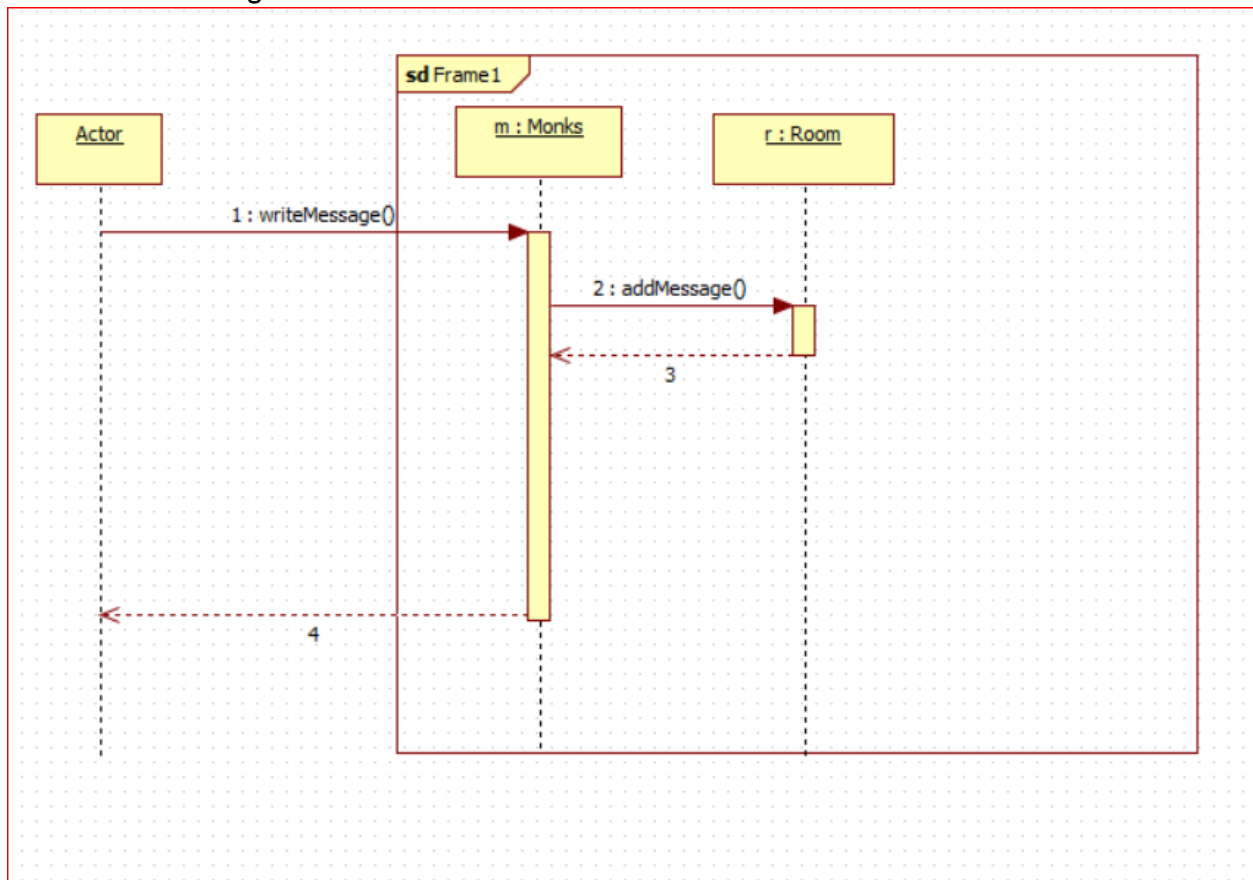
5.1.3 Self Kill



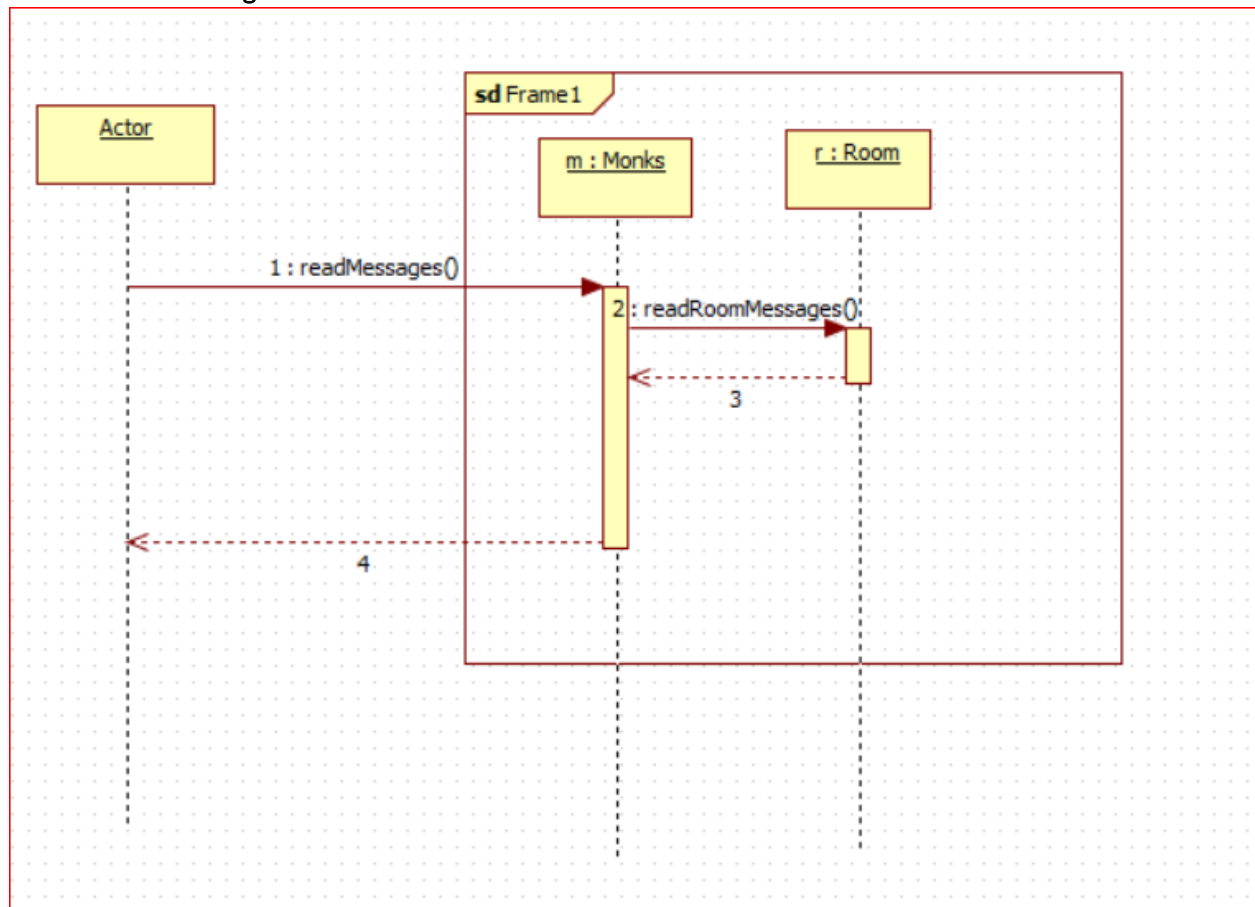
5.1.4 Daemon action



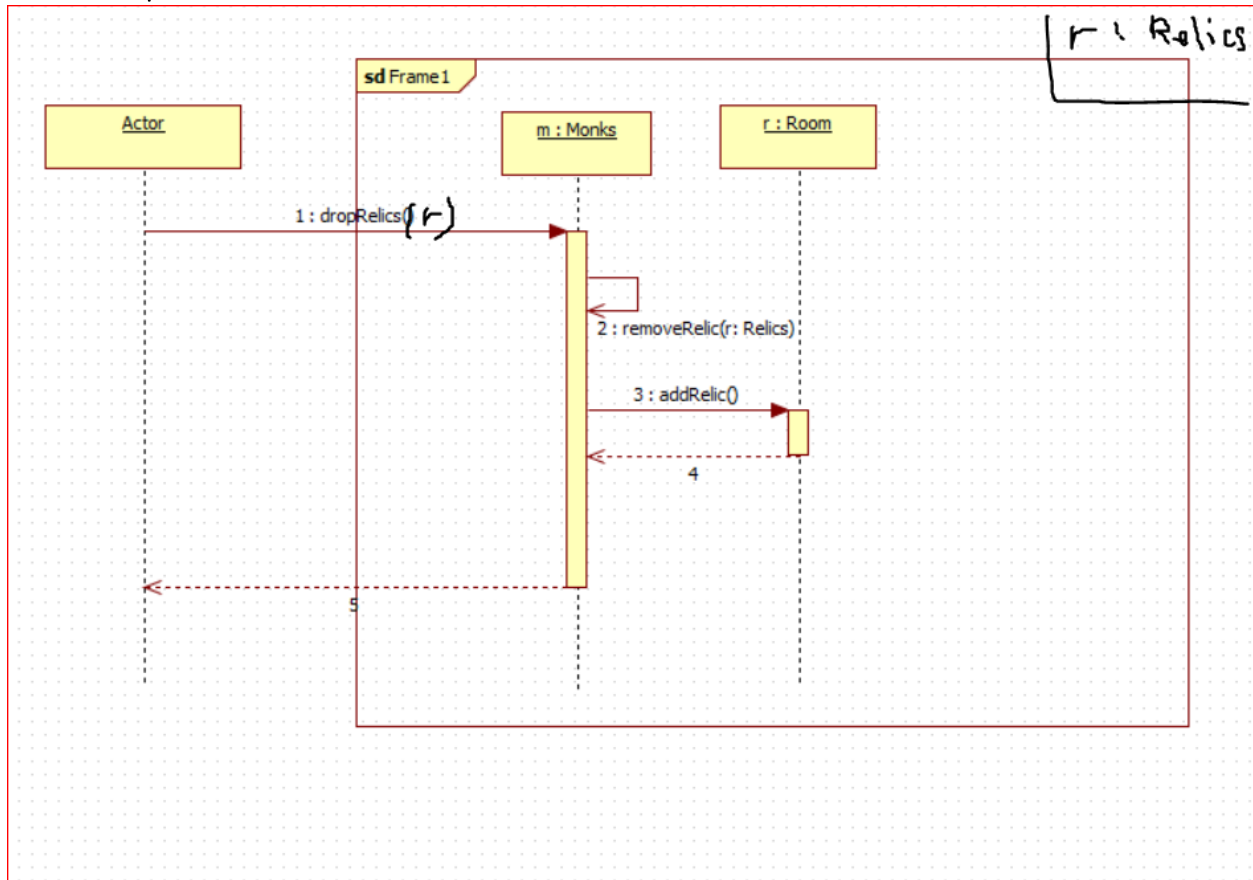
5.1.5 Write message



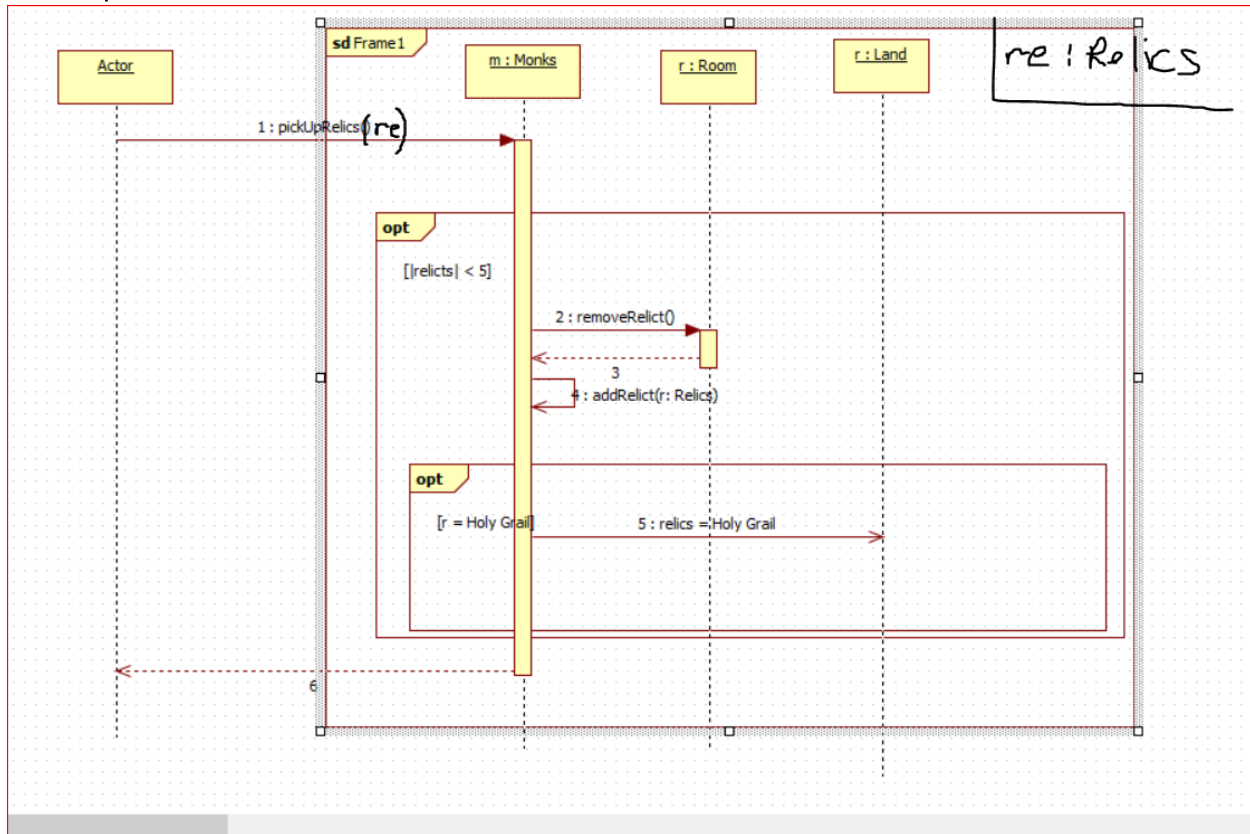
5.1.6 Read messages



5.1.7 Drop relic



5.1.8 pick UP Relic

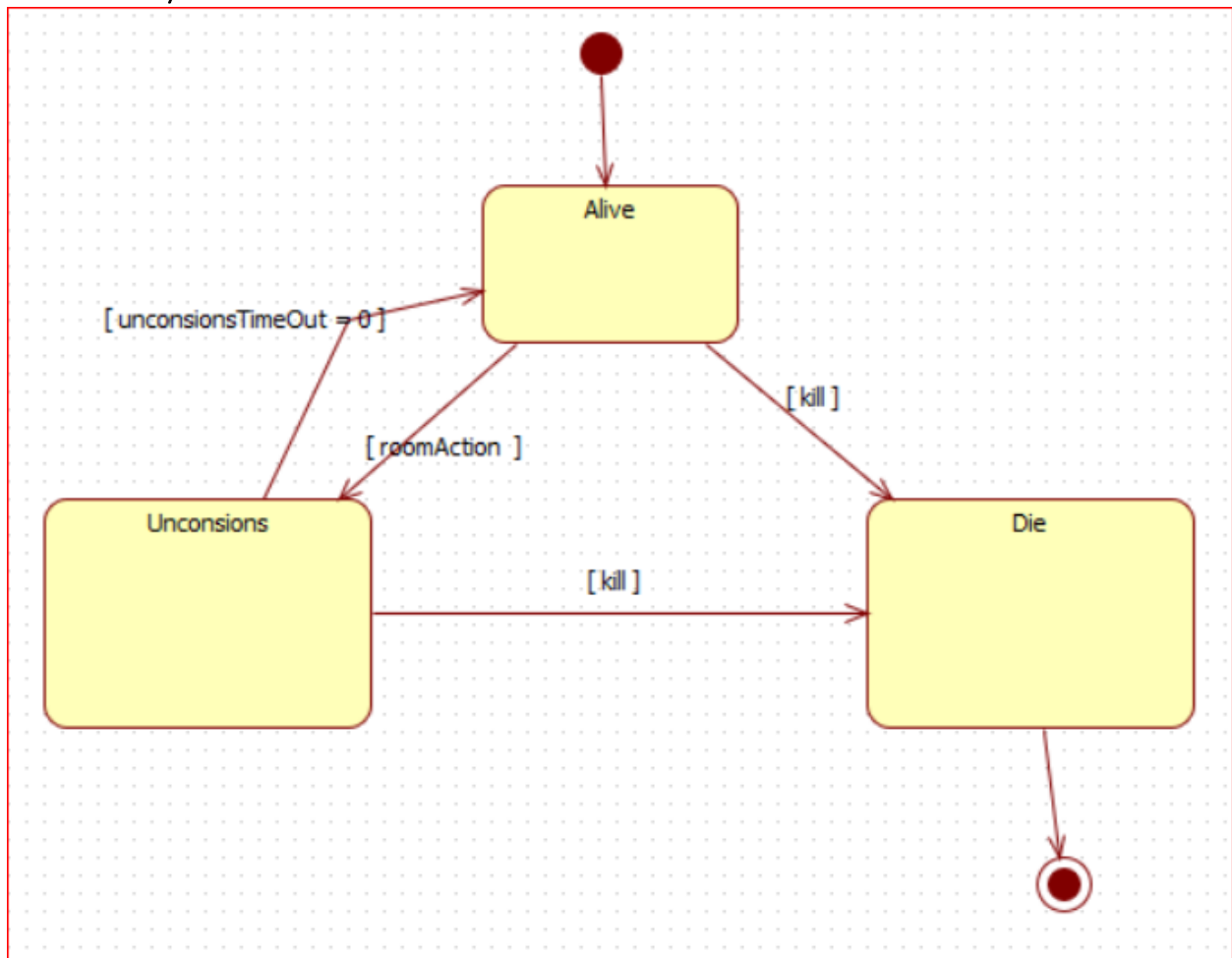


I put important that all objects is alive in all time for all game, and just switch them states.

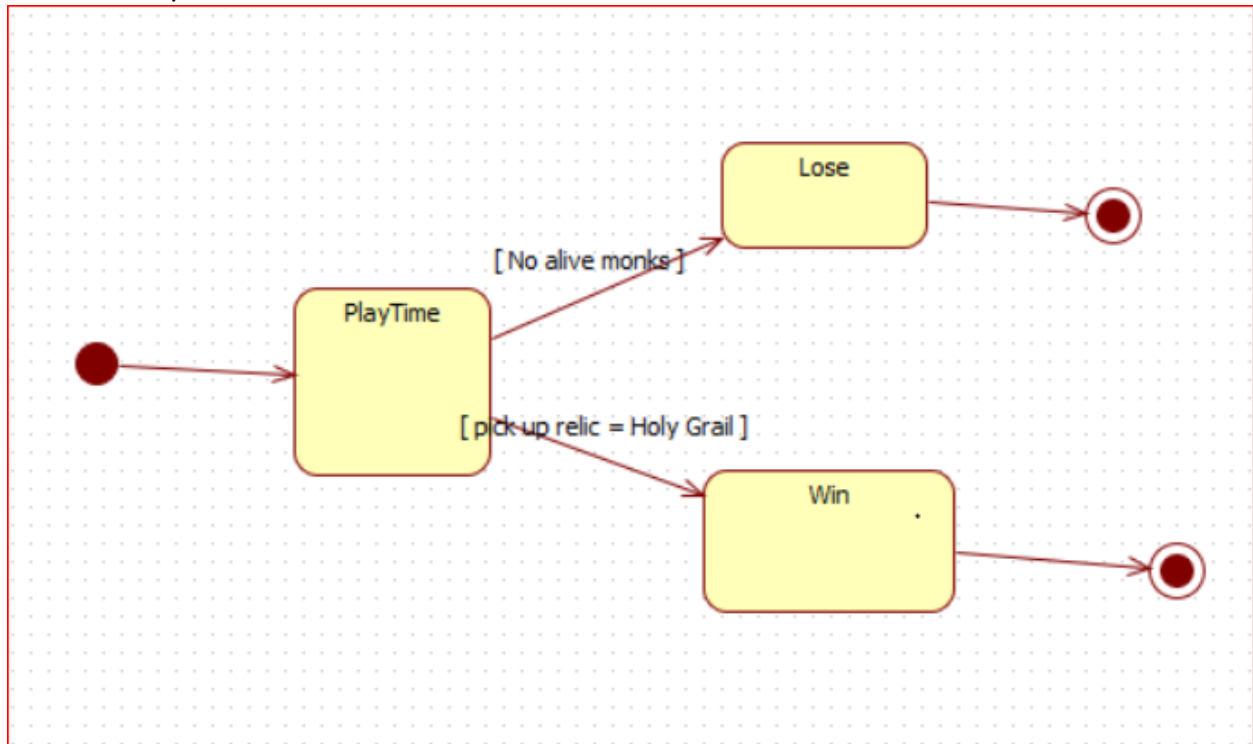
ALSO I PUT ARGUMENTS BY HAND WHERE WHITESTAR UML WOULDN'T LIKE TO DO IT, AND COMMENT WHICH CLASS I MEAN BY THAT IN THE CORNER

5.2 STATE MACHINE DIAGRAMS

5.2.1 Live cycle of monks



5.2.2 Win cycle



By cycle I mean – process (states)

6 WORK ACTIVITY LOG

Start	Duration	Work performed	References
2023.01.24	6 hours	Prepared by hands all diagrams and start describe	
2023.01.25	3 hours	continue with class diagram and decribing	
2023.01.26	10 hours	Sequce and all parts solve	

Total work activity: 19 hours

Modeling tool: WhiteStarUML

Other tools: Docx, bme.iit, <https://www.uml-diagrams.org/>