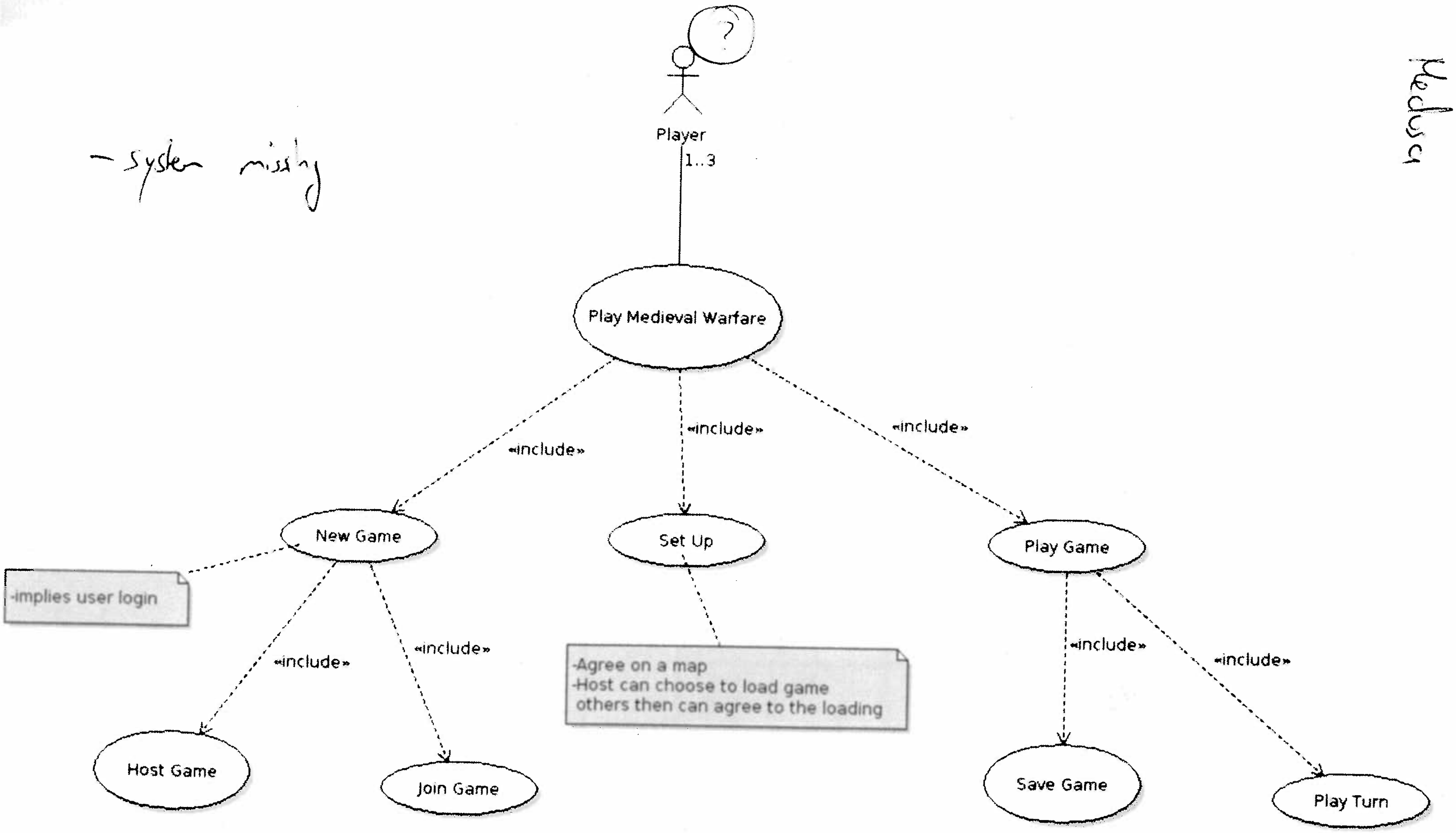
- system missing

Medusa



always the same

| | |
|---|---|
| **Use Case** | Play Medieval Warfare |
| **Scope** | all |
| **Level** | Summary |
| **Intention in Context** | The intention of the Player is to be able to achieve all different requirements in order to finally play the game. This doesn't include only playing itself but it includes all setup and hosting stuff<br>? |
| **Multiplicity** | 4 |
| **Primary Actor** | Player |
| **Main Success Scenario** | 1. Player starts a new game, completes logging (either by joining one, or creating and hosting one)<br>2. Player setup all that is needed (agree on a map, possibly load a game)<br>3. Player plays a turn<br>4. Player finishes or saves the game |
| | |
| **Extensions** | 2a. Fails if Player tries to load a game but chosed to join an existing game, then must select new game back at step 1 |

| | |
|---|---|
| **Use Case :** | New Game |
| **Scope :** | Server, UI, Player |
| **Level :** | User Goal |
| **Intention in Context :** | The intention of the Player is to be able to fill in the logging info, and decide to start a new game or join an existing game |
| **Multiplicity :** | 1 |
| **Primary Actor :** | Player |
| **Main Success Scenario :** | 1. Player fills in the logging<br>2. Player decide to host or join a game |
| | |
| **Extensions :** | ? |

?

| | |
|---|---|
| **Use Case :** | Set Up |
| **Scope :** | Engine, Player,Server |
| **Level :** | User Goal |
| **Intention in Context :** | The Player intention is to setup the remaining requirements in order to be ready to play. (Agree on a map, load a previously saved game, etc.) |
| **Multiplicity :** | 4 |
| **Primary Actor :** | Player |
| **Main Success Scenario :** | 1. Load a map (new or existing one) |
| | 2. Agree on the map with other players |
| | 3. Require all 4 players to be ready |
| **Extensions :** | 3a. If not enough players, then wait for players or try another map which might have enough player waiting |

| | |
|---|---|
| **Use Case :** | Play Game |
| **Scope :** | Engine, UI |
| **Level :** | User Goal |
| **Intention in Context :** | The Player intention is to start playing the game which means exchanging turns between players, and save the game when you want to stop playing |
| **Multiplicity :** | 4 |
| **Primary Actor :** | Player |
| **Main Success Scenario :** | 1. Play you turn |
| | 2. Look at other players turns |
| | 3. Save the game |
| **Extensions :** | 2a. If other players stop playing, then game ends |

← looking at play turn use case this also includes other players turn

not all inputs and outputs are shown

- can a player play more than 1 turn?

| | |
|---|---|
| **Use Case :** | Save Game |
| **Scope :** | Engine, Map |
| **Level :** | Subfunction |
| **Intention in Context :** | The intention of the Player is the save all the data associated with the current game in order to play able to play at a later time |
| **Multiplicity :** | 4 |
| **Primary Actor :** | Player |
| **Main Success Scenario :** | 1. Stop Playing |
| | 2. Save the game |
| | |
| **Extensions :** | |

| | |
|---|---|
| **Use Case :** | Play Turn |
| **Scope :** | Engine, UI, Map, Unit |
| **Level :** | Subfunction |
| **Intention in Context :** | The intention of the Player is to do some moves in order to play the game and then control is given over to other players until its his turn again |
| **Multiplicity :** | 4 |
| **Primary Actor :** | Player |
| **Main Success Scenario :** | 1. Player does moves |
| | 2. Player hands control over to other players |
| | 3. Player regains control |
| **Extensions :** | |

inpts ?

otpts ?

| | |
|---|---|
| **Use Case :** | Host Game |
| **Scope :** | Server, Player |
| **Level :** | Subfunction |
| **Intention in Context :** | The intention of the Player is to start a completely new game. It then becomes the host of this game, so he is the one who can take decision about setup |
| **Multiplicity :** | 1 |
| **Primary Actor :** | Player |
| **Main Success Scenario :** | 1. Player select New Game |
| | 2. Then, Player select Host Game |
| | |
| **Extensions** | |

| | |
|---|---|
| **Use Case :** | Join Game |
| **Scope :** | Server, UI |
| **Level :** | Subfunction |
| **Intention in Context :** | The Player intention is to join a lobby that another host created and a host decide on game and play with them |
| **Multiplicity :** | 3 |
| **Primary Actor :** | Player |
| **Main Success Scenario :** | 1. Look at existing games |
| | 2. Join your prefered existing game |
| | |
| **Extensions :** | 1a. If no existing game present in lobby, then Player must create a new game |

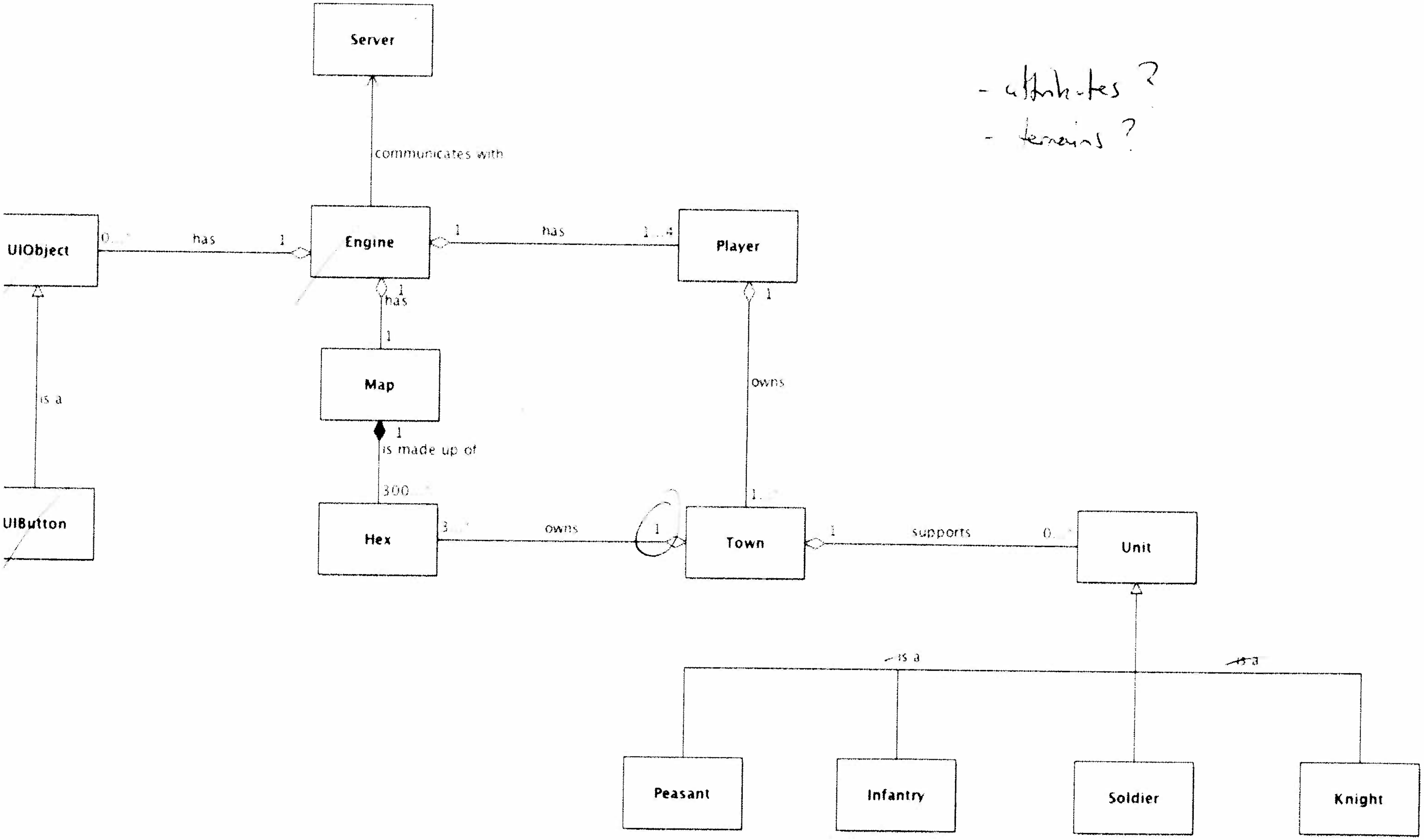
System presents Player with a list of existing games.

Player informs System about the game she wants to join

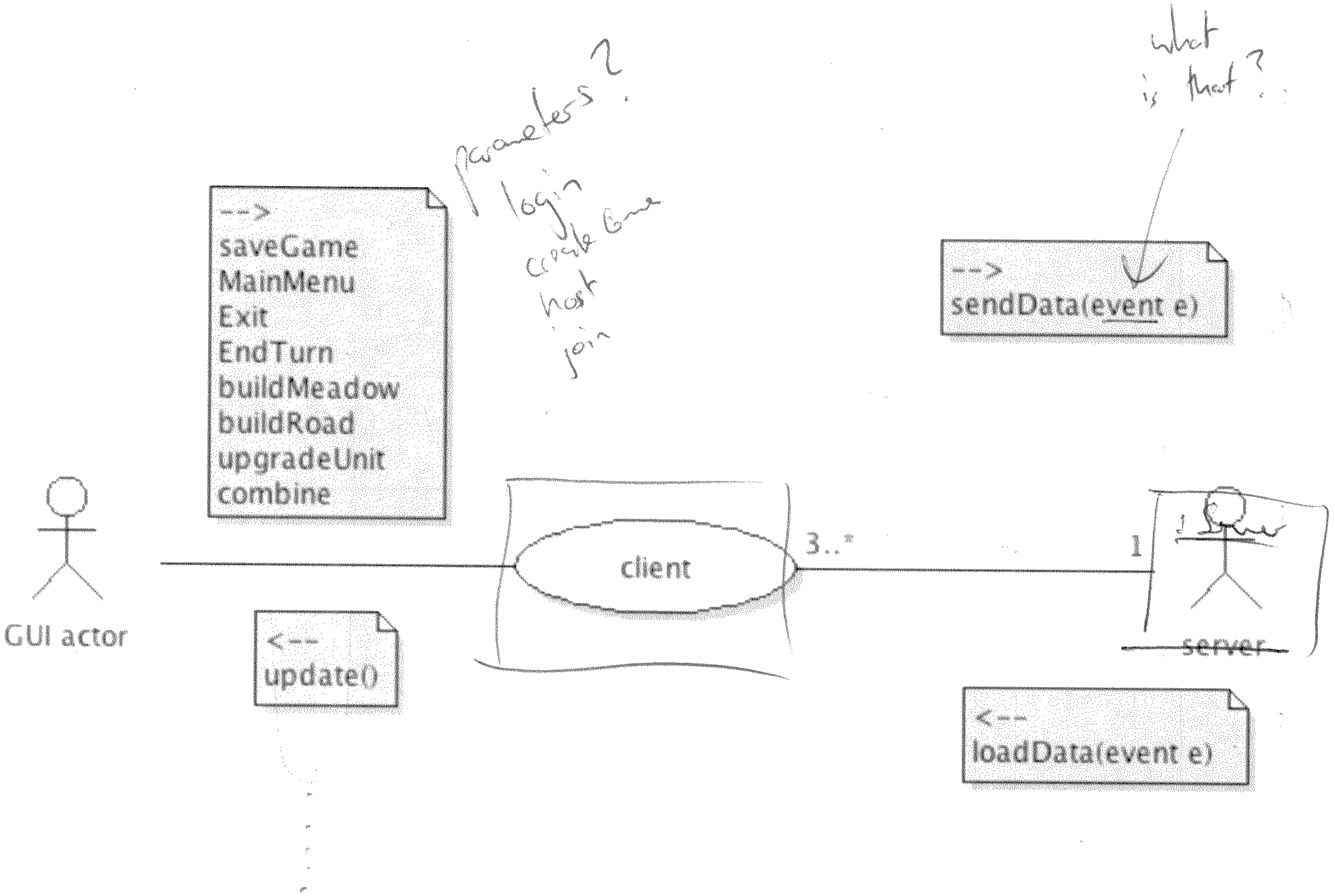
1a. There are currently no games in progress.

1a.1. System notifies Player that there are no games in progress and that he must create a new game. Use case ends in failure

Of the client, I presume.



- attributes ?
- terrains ?


-->
saveGame
MainMenu
Exit
EndTurn
buildMeadow
buildRoad
upgradeUnit
combine
parameters?
login
create Game
host
join
what is that?
-->
sendData(event e)
GUI actor
client
3..*
1
1 Server
~~server~~
<--
update()
<--
loadData(event e)

How do you know which unit is to be upgrade ?

**Type Definitions**

type Unit = enum {peasant , Infantry ...

tombstone}

**Input Messages**

LoadUserStats ( userName : String )

CreateNewUser ( )

InitializeGame ( )

DistributeMap - host ( )

AgreeOnMap- join ( )

UpdateUnitChange ( unitType : Unit , amount : Integer )

SaveGame ( )

UpdatePlayerStats ( userName : String , statistics : Int )

**utput Messages**

EnterUserName (Scanner)

CreateUserName (Scanner)

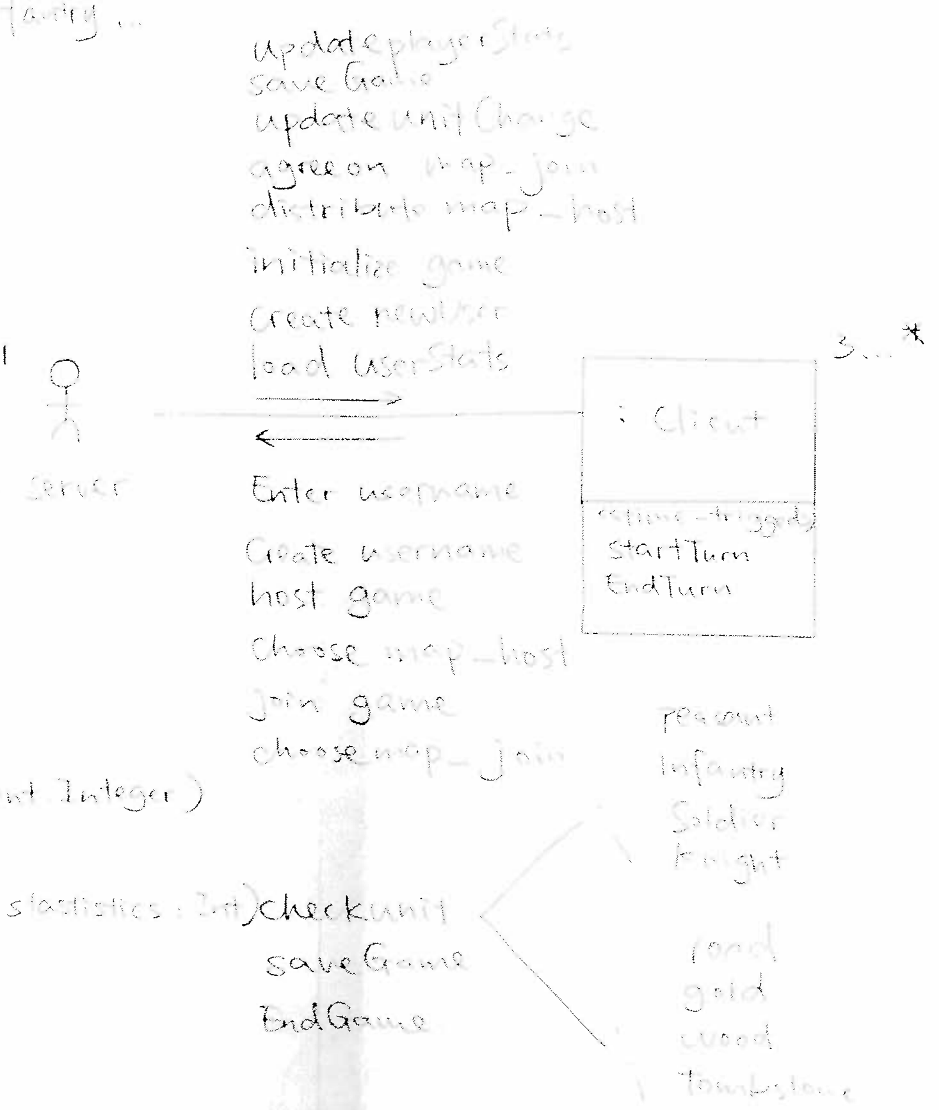
HostGame ( )

Choose Map - host ( )

JoinGame ( )

ChooseMap- join ( )

CheckUnit ( unitType : unit , state : boolean ).



client

| | |
|---|---|
| **Operation :** | ~~GUIactor~~ :: saveGame |
| **Scope :** | engine |
| **Messages :** | |
| **New :** | GeneratedGameData file |
| **Pre :** | click Menu |
| **Post :** | Write to a file current game data |
| **Use Cases :** | saveGame use case |

| | |
|---|---|
| **Operation :** | GUI actor :: mainMenu |
| **Scope :** | UIObject |
| **Messages :** | |
| **New :** | |
| **Pre :** | When being in game |
| **Post :** | Clear the game data and brings you back to lobby to start a new game |
| **Use Cases :** | |

| | |
|---|---|
| **Operation :** | GUI actor :: upgradeUnit |
| **Scope :** | unit |
| **Messages :** | Unit :upgrade |
| **New :** | upgradedUnit |
| **Pre :** | Enough money in the city, must not already be a knight |
| **Post :** | destroy current unit and creates upgraded unit |
| **Use Cases :** | |

which one?

where is it created?

decrease money of city

| | |
|---|---|
| **Operation :** | GUI actor :: combine |
| **Scope :** | unit |
| **Messages :** | unit : combined units |
| **New :** | upgradedUnit |
| **Pre :** | need the require amount of unit (two peasants, … |
| **Post :** | previous units deleted, new stronger unit created |
| **Use Cases :** | |

| | |
|---|---|
| **Operation :** | GUI actor :: buildRoad |
| **Scope :** | hex, unit, map |
| **Messages :** | Unit :: {HexBeingBuilt} |
| **New :** | |
| **Pre :** | Hex must be empty (no forest on it) |
| **Post :** | Builds of a road (one turn required to complete) |
| **Use Cases :** | |

| | |
|---|---|
| **Operation :** | GUI actor :: buildMeadow |
| **Scope :** | hex, unit, map |
| **Messages :** | Unit :: {HexBeingBuilt} |
| **New :** | |
| **Pre :** | Hex must be empty (no forest on it) |
| **Post :** | Initiates the building of a meadow (two turns required to complete) |
| **Use Cases :** | |

on model

| | |
|---|---|
| **Operation :** | GUI actor :: exit |
| **Scope :** | engine |
| **Messages :** | |
| **New :** | |
| **Pre :** | When anywhere and you want to kill everything |
| **Post :** | Whole game closed, back to desktop |
| **Use Cases :** | |

| | |
|---|---|
| **Operation :** | GUI actor :: endTurn |
| **Scope :** | engine |
| **Messages :** | Player : TakeControl |
| **New :** | ActionEventsBundle |
| **Pre :** | You have to be the current player in order to end your turn |
| **Post :** | Control shifted over to other player |
| **Use Cases :** | |

| | |
|---|---|
| **Operation :** | Client :: sendData |
| **Scope :** | server |
| **Messages :** | |
| **New :** | Client : new data event send to server |
| **Pre :** | |
| **Post :** | Sends request to server and server will unpack it |
| **Use Cases :** | |

| | |
|---|---|
| **Operation :** | Client :: update |
| **Scope :** | engine |
| **Messages :** | |
| **New :** | |
| **Pre :** | |
| **Post :** | observer callback to tell GUI to regenerate data |
| **Use Cases :** | |

| | |
|---|---|
| **Operation :** | Server :: loadData |
| **Scope :** | server |
| **Messages :** | |
| **New :** | Server :: dataUnit |
| **Pre :** | Data must have been previously saved |
| **Post :** | Data from the server uploaded |
| **Use Cases :** | |