Sequence Diagram

Participants: Actor, MowersSystem, Mower, Gopher

- Actor → MowersSystem: **upload starting file**

Alternative

[file is invalid]

Text

- MowersSystem ⇠ Actor: **ERROR: Test scenario file name not found**

[Else]

- MowersSystem ⇠ Actor: **initilize successfylly**

- Actor → MowersSystem: **start system simulation**

Loop

[for each simulation turn]

Alternative

[all mowers crash || all grass mow]

- Mower → MowersSystem: **halt simulation**
- MowersSystem → Actor: **fianl report**

[Else]

Loop

[for each mower]

Alternative

[mower was crash || 0 energy]

continue

[Else]

- MowersSystem → Mower: **poll Mower For Action base on strategy**
  - **mower comunication**
- Mower ⇠ MowersSystem: **track action status**
- MowersSystem → Mower: **validate Mower action**

Alternative

[action: pass]

- Mower ⇠ MowersSystem: **report mower status**
- MowersSystem → Actor: **report mower ID action and status**

[action: sscan]

- Mower ⇠ MowersSystem: **sscan result**
- MowersSystem → Actor: **report mower ID, action and sccan result**

[action: lscan]

- Mower ⇠ MowersSystem: **lscan result**
- MowersSystem → Actor: **report mower ID, action and scoan result**

[action: steer]

- Mower ⇠ MowersSystem: **report mower direction**
- Mower ⇠ MowersSystem: **report mower status**
- MowersSystem → Actor: **report mower ID, action, direction and status**

[action: move]

- Mower ⇠ MowersSystem: **report mower status**
- **update lawn info, cut grass number if mow**
- MowersSystem → Actor: **report mower ID, action, and status**

Loop

[for each gopher && turn % gopher movement period == 0]

- MowersSystem → Gopher: **poll each gopher to action**
  - **select mower to chase**

Altnative

[mower and gophor in same squre]

- Gopher → Mower: **chew up mower**
- Mower → MowersSystem: **update mower status and lawn info**

- Gopher → MowersSystem: **update gopher location and status**

Text

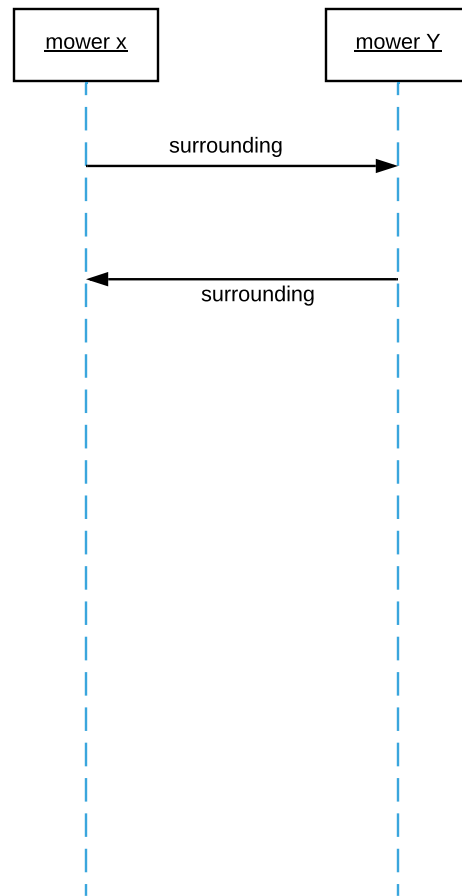- MowersSystem → Actor: **Gophoer ID, selected mower ID, location and status**

mower comunication:

select mower to chase:

mower x   mower Y

gopher x   MowerSystem   gopher Y

surrounding

BFS: nearest mower id x

BFS:nearest mower id y where y!=x

surrounding

<gopherx>,<mower x>,location

status: ok

<gopherY>,<mower y>,location

status: ok

Actor   **MowersSystem**   **Mower**   **Gopher**

Actor

upload starting file

Alternative

[file is invalid]

Text

ERROR: Test scenario file name not found

[Else]

initilize successfylly

start system simulation

Loop

[for each simulation turn]

Alternative

[all mowers crash || all grass mow]

halt simulation

fianl report

[Else]

Loop

[for each mower]

Alternative

[mower was crash || 0 energy]

continue

Loop

[for each mower]

Alternative

[mower was crash || 0 energy]

continue

[Else]

poll Mower For Action base on strategy

mower comunication

track action status

validate Mower action

Alternative

[action: pass]   report mower status

report mower ID action and status

[action: sscan]

sscan result

report mower ID, action and sccan result

[action: lscan]

lscan result

report mower ID, action and sccan result

[action: steer]

report mower direction

report mower status

report mower ID, action, direction and status

[action: move]   report mower status

update lawn info, cut grass number if mow

report mower ID, action, and status

report mower ID, action and scan result

[action: steer]

report mower direction

report mower status

report mower ID, action, direction and status

[action: move]   report mower status

report mower ID, action, and status   update lawn info, cut grass number if mow.

Loop

[for each gopher && turn % gopher movement period == 0]

poll each gopher to action

select mower to chase

Altnative

[mower and gophor in same squre]   chew up mower

update mower status and lawn info

update gopher location and status

Text

Gophoer ID, selected mower ID, location and status