UML Seq Farmer and Food
(only Farming)

Chantelle Loh Yi Wei

Participants: :Farmer, :SowCropBehaviour, :FertilizeBehaviour, :HarvestingBehaviour, :GameMap, :Exit, :SowCropAction, :FertilizeAction, :Crops, :Food

Loop [Iterate list of behaviour]

:Farmer → :SowCropBehaviour : call

:SowCropBehaviour → :GameMap : get locationOf actor

Alternative [if Ground.getDisplayChar =='.']
:GameMap → :Exit : check correctGround()
:Exit --> :GameMap : exit is suitable
:GameMap --> :SowCropBehaviour : Use this exit if have
[Else]
:Exit --> :GameMap : no suitable exit

:SowCropBehaviour → :SowCropAction : at that exit location , return SowCropAction()
:SowCropAction → :Crops : add Crops object

:Farmer → :FertilizeBehaviour : call
:FertilizeBehaviour → :GameMap : get map.locationOf(actor)

Alternative [if Item.getDisplayChar =='c']
:GameMap → :Exit : check correctItem()
:Exit --> :GameMap : exit is suitable
:GameMap --> :FertilizeBehaviour : Use this exit
[Else]
:Exit --> :GameMap : no suitable exit

> **Note:** this 2 alternative is implemented by creating an abstract class claled GrowingBehaviour which have a method for this as both of them involve checking the item.getDiaplayChar.

:FertilizeBehaviour → :FertilizeAction : return FertilizeAction()
:FertilizeAction → :Crops : fertilise()

:Farmer → :HarvestingBehaviour : call
:HarvestingBehaviour → :GameMap : get map.locationOf(actor)

Alternative [if Item.getDisplayChar =='C']
:GameMap → :Exit : check correctItem()
:Exit --> :GameMap : exit is suitable
:GameMap --> :HarvestingBehaviour : Use this exit
[Else]
:Exit --> :GameMap : no suitable exit

:HarvestingBehaviour → :Food : Add food object and remove crop