

Entity

Food

Foodtype (Enum)	
	amount
raspberry	1
apple	2
pineapple	3

Gatherer

State (Enum)							
idle	following	wandering	charging	fleeing	collecting	beaten	exhausted

	arguments	description
follow	entity Entity	starts to follow entity
attackmove	gatherer Gatherer	starts to follow gatherer, attacks when close, steals food on successful attack
collect	food Food	starts to move towards food, collects when close
wander	count(optional) int or float	starts to wander randomly, optionally you can pass "approximate number of direction changes per 10 secs"
escape	entity Entity	starts to move in opposite direction of entity
cancel		cancels active task

Gatherer Abilities (can be used multiple times each step)

function	arguments	return	description												
self.assignTask(task, taskarg)	<table><tr><td>task</td><td>Task</td></tr><tr><td>taskarg</td><td>Task/arguments</td></tr></table>	task	Task	taskarg	Task/arguments		Assigns task, most recent task assigned will be executed in next step								
task	Task														
taskarg	Task/arguments														
self.getdistance(entity)	<table><tr><td>entity</td><td>Entity</td></tr></table>	entity	Entity	distance(float)											
entity	Entity														
self.getdirection(entity)	<table><tr><td>entity</td><td>Entity</td></tr></table>	entity	Entity	angle(float)	returns relative direction angle for an entity's position such that <table><tr><td>angle</td><td>direction</td></tr><tr><td>0.0</td><td>north</td></tr><tr><td>90.0</td><td>east</td></tr><tr><td>±180.0</td><td>south</td></tr><tr><td>-90.0</td><td>west</td></tr></table>	angle	direction	0.0	north	90.0	east	±180.0	south	-90.0	west
entity	Entity														
angle	direction														
0.0	north														
90.0	east														
±180.0	south														
-90.0	west														
self.getfacing(gatherer)	<table><tr><td>gatherer</td><td>Gatherer</td></tr></table>	gatherer	Gatherer	angle(float)	returns facing angle such that if 0.0 other gatherer is directly facing, if 180 other gatherer is facing back (if 0.0 other gatherer may be charging at you!!)										
gatherer	Gatherer														
self.closestfood(foodtype=None)	<table><tr><td>foodtype(optional)</td><td>Foodtype</td></tr></table>	foodtype(optional)	Foodtype	Food	closest known food, None if no food known										
foodtype(optional)	Foodtype														
self.visiblefoods(foodtype=None)	<table><tr><td>foodtype(optional)</td><td>Foodtype</td></tr></table>	foodtype(optional)	Foodtype	list(Food)	lists sorted wrt distance (closest has lowest index), None if no food visible										
foodtype(optional)	Foodtype														
self.knownfoods(foodtype=None)	<table><tr><td>foodtype(optional)</td><td>Foodtype</td></tr></table>	foodtype(optional)	Foodtype	list(Food)	lists sorted wrt distance (closest has lowest index), None if no food known. Not all that is known is unpicked..										
foodtype(optional)	Foodtype														
self.isvisible(food)	<table><tr><td>food</td><td>Food</td></tr></table>	food	Food	bool	if not visible, a known food may have already been collected!										
food	Food														
self.closestgatherer()		Gatherer													
self.visiblegatherers()		list(Gatherer)	lists sorted wrt distance (closest has lowest index)												
self.foodcarried(gatherer)	<table><tr><td>gatherer</td><td>Gatherer</td></tr></table>	gatherer	Gatherer	(int)	returns how fat is the gatherer's backpack, can pass "self" <table><tr><td>%0-%25</td><td>0</td></tr><tr><td>%25-%50</td><td>1</td></tr><tr><td>%50-%75</td><td>2</td></tr><tr><td>%75-%100</td><td>3</td></tr></table>	%0-%25	0	%25-%50	1	%50-%75	2	%75-%100	3		
gatherer	Gatherer														
%0-%25	0														
%25-%50	1														
%50-%75	2														
%75-%100	3														
self.readytoattack()			returns True if attack is not in cooldown, else return False												
self.checkstate(gatherer)	<table><tr><td>gatherer</td><td>Gatherer</td></tr></table>	gatherer	Gatherer	state(State)	returns current state of a gatherer, can pass "self"										
gatherer	Gatherer														
self.checkfoodtype(food)	<table><tr><td>food</td><td>Food</td></tr></table>	food	Food	foodtype(Foodtype)	returns enumerated foodtype e.g Foodtype. berry										
food	Food														