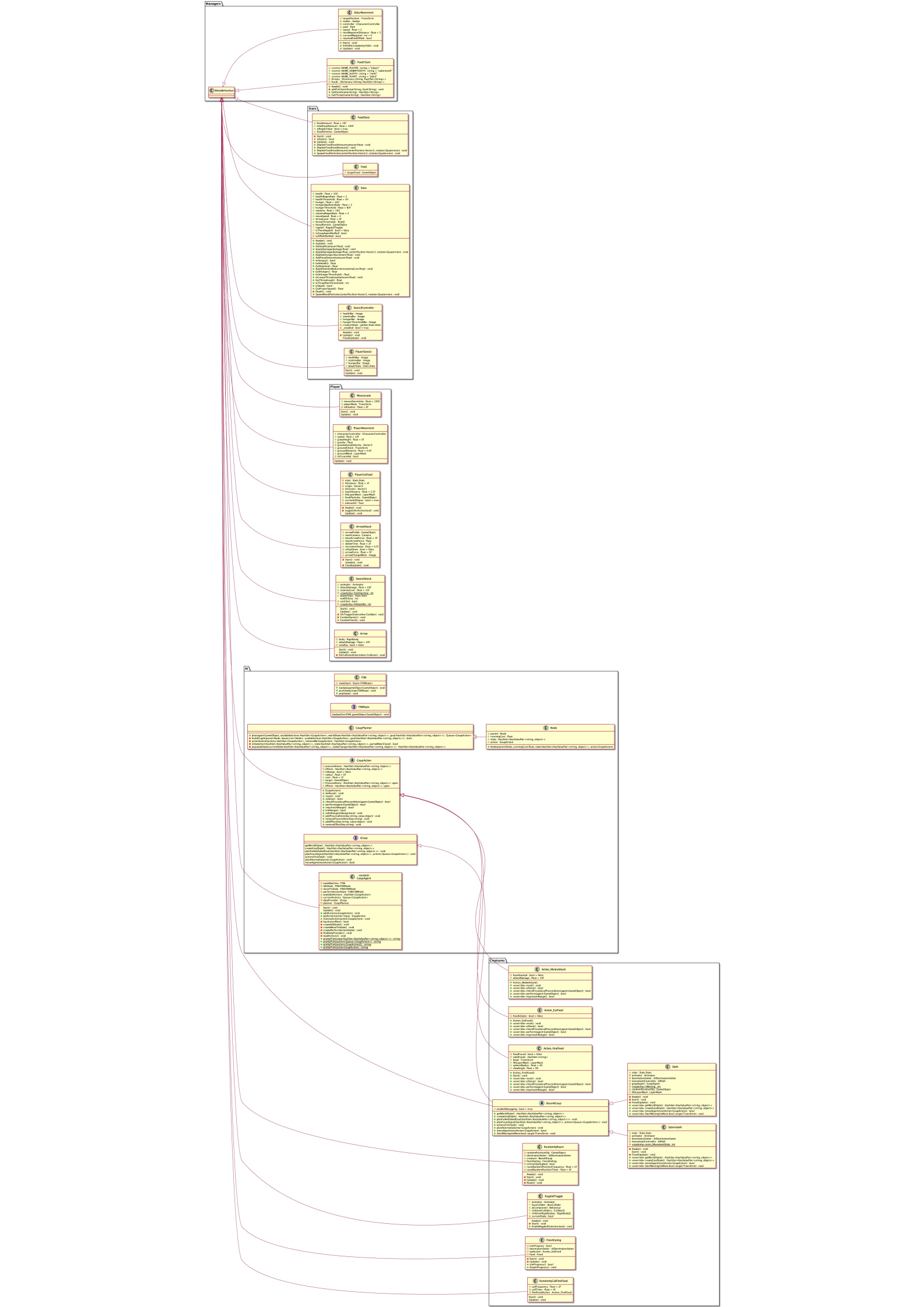
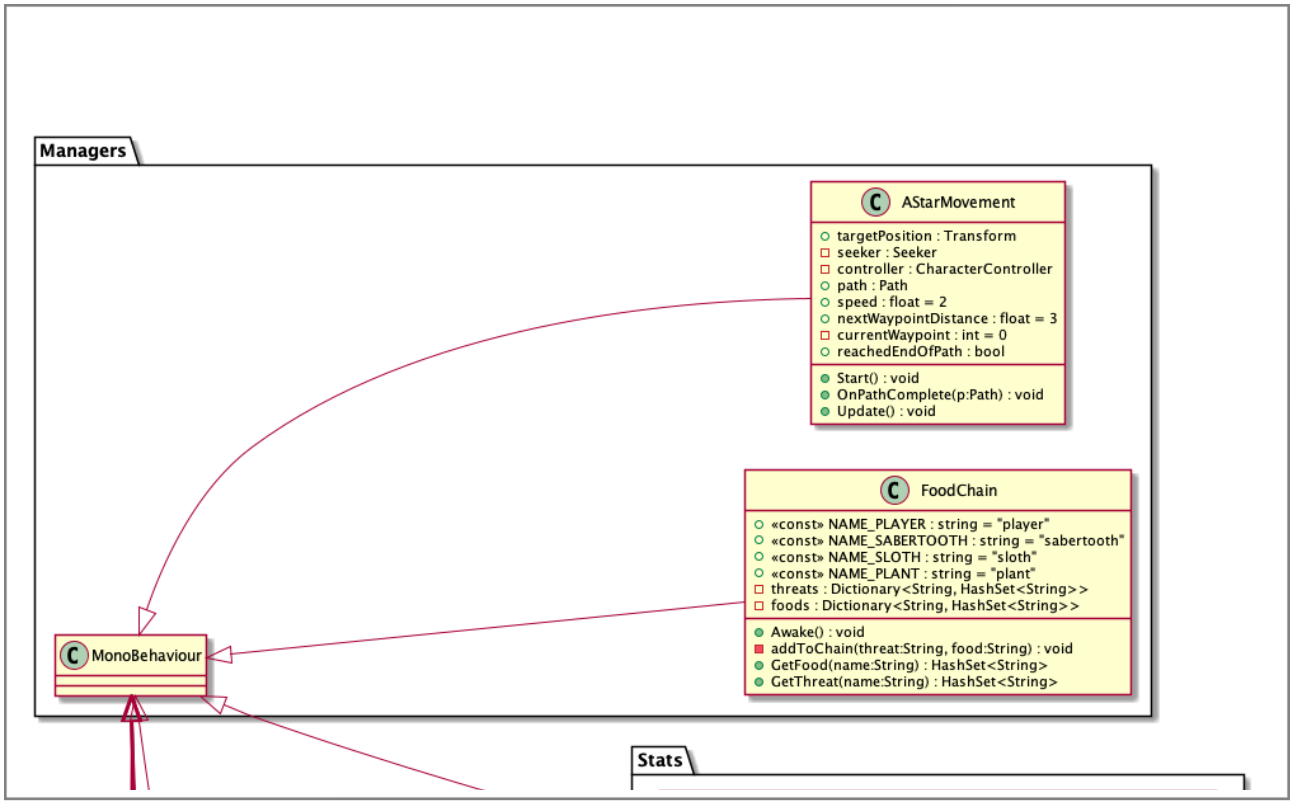
Appendix E

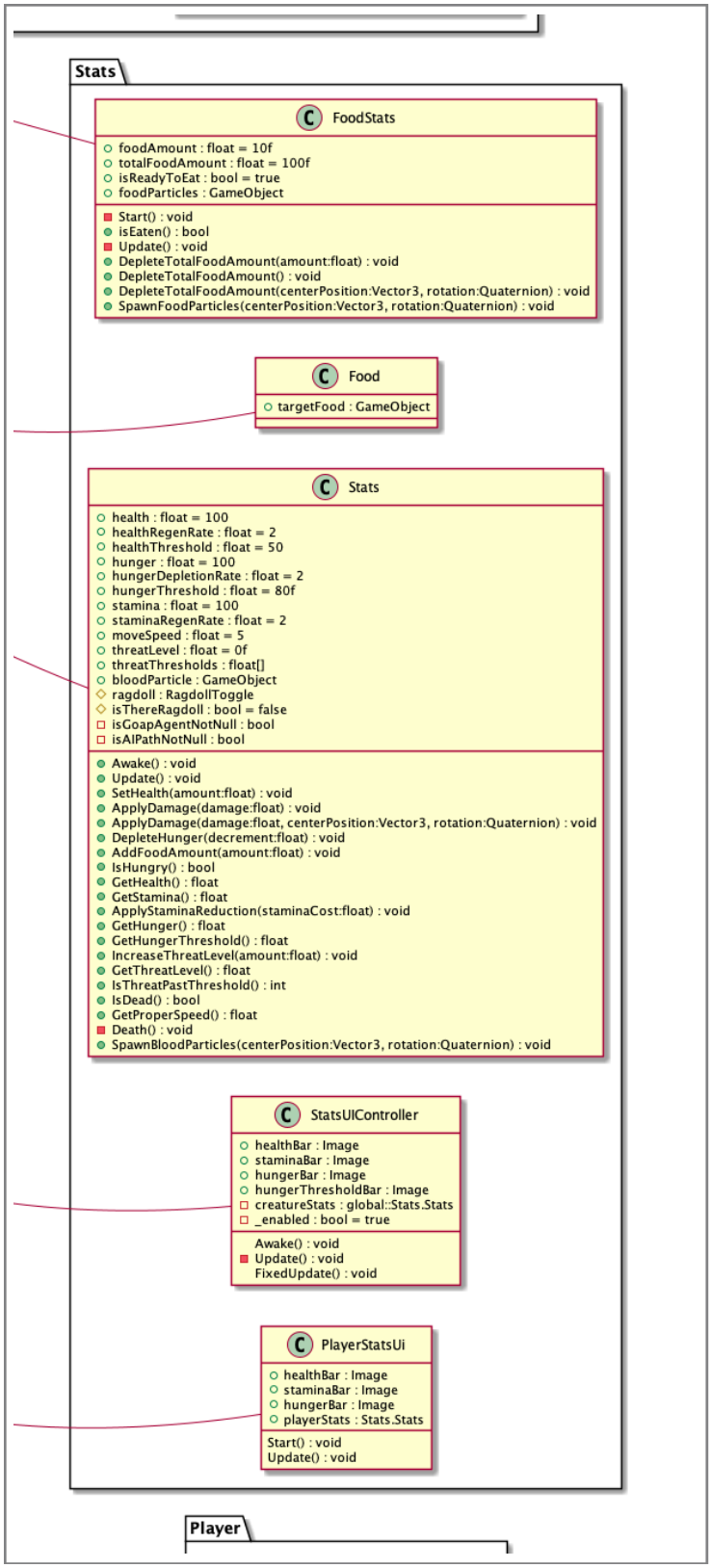
Final Class Diagram


Managers
AlterMovement
- targetPosition : Transform
- agent : NavMeshAgent
- controller : CharacterController
- path : Path
- speed : float = 2
- nextWaypointDistance : float = 3
- reachedEndOfPath : bool
- reachedTestPath : bool
+ Start() : void
+ CreatePath(completeCallback) : void
+ Update() : void
FoodChain
- <const> NAME_PLAYER : string = "player"
- <const> NAME_SABRETOOTH : string = "sabertooth"
- <const> NAME_SLOTH : string = "sloth"
- <const> NAME_MAMMOTH : string = "mammoth"
- threats : Dictionary<string, HashSet<string>>
- foods : Dictionary<string, HashSet<string>>
+ Awake() : void
+ SetEatFood(creatureString: string, foodString) : void
+ GetFoodsName(string) : HashSet<string>
+ GetThreatsName(string) : HashSet<string>
MonoBehaviour
Stats
FoodStats
- foodAmount : float = 100
- isFoodDepleted : bool = false
- isReadyToEat : bool = true
- foodParticles : GameObject
+ Start() : void
+ IsEaten() : bool
+ IsReadyToEat() : bool
+ DepleteFoodByFoodAmount(amountToFeed) : void
+ DepleteFoodByFoodAmount() : void
+ SpawnFoodParticles(centerPosition:Vector3, rotation:Quaternion) : void
+ SpawnFoodParticles(centerPosition:Vector3, rotation:Quaternion) : void
Food
- targetFood : GameObject
Stats
- health : float = 100
- healthRegenRate : float = 2
- maxHealth : float = 100
- hunger : float = 100
- hungerDepletionRate : float = 2
- maxHunger : float = 100
- stamina : float = 100
- staminaRegenRate : float = 2
- maxStamina : float = 100
- thirstLevel : float = 0f
- thirstThresholds : float[]
- ragdollToggle : RagdollToggle
- ragdoll : RagdollToggle
- isTherePlayDoll : bool = false
- isTherePlayerDoll : bool
- isChangeHealth : bool
+ Awake() : void
+ Update() : void
+ SetHealthAmount(float) : void
+ ApplyDamage(damage:float) : void
+ ApplyDamage(damage:float, centerPosition:Vector3, rotation:Quaternion) : void
+ DepleteHunger(decrement:float) : void
+ AddFood(increaseHunger:float) : void
+ IsDead() : bool
+ GetHealth() : float
+ GetStamina() : float
+ ApplyStaminaReduction(staminaCost:float) : void
+ IsStaminaLow() : bool
+ GetHungerThreshold() : float
+ IncreaseThirstLevel(amount:float) : void
+ ResetThirstLevel() : void
+ IsThreePlayDollThreshold() : int
+ IsDead() : bool
+ GetHealthRatio() : float
+ Die(int) : void
+ SpawnBloodParticles(centerPosition:Vector3, rotation:Quaternion) : void
StatsUIController
- healthBar : Image
- staminaBar : Image
- hungerBar : Image
- thirstBar : UnityEngine.Image
- creatureStats : global::Stats.Stats
- _visibled : bool = true
Awake() : void
Update() : void
FixedUpdate() : void
PlayerStatsUI
- healthBar : Image
- staminaBar : Image
- hungerBar : Image
- playerStats : Stats.Stats
Start() : void
Update() : void
Player
MouseLook
- mouseSensitivity : float = 100f
- playerBody : Transform
- xRotation : float = 0f
Start() : void
Update() : void
PlayerMovement
- characterController : CharacterController
- speed : float = 12
- jumpHeight : float = 3f
- gravity : float
- velocity : UnityEngine.Vector3
- groundCheck : Transform
- groundDistance : float = 0.4f
- groundMask : LayerMask
- isGrounded : bool
Update() : void
PlayerEatFood
- stats : Stats.Stats
- thickness : float = 1f
- origin : Vector3
- direction : Vector3
- maxDistance : float = 2.5f
- hitLayerMask : LayerMask
- foodParticles : GameObject
- isEatingAnimation : bool = true
- interactUI : Text
+ Awake() : void
+ ToggleUIActive(bool) : void
Update() : void
ArrowAttack
- arrowPrefab : GameObject
- mainCamera : Camera
- baseArrowForce : float = 9f
- maxArrowForce : float = 27
- deleteTime : float = 2f
- incrementValue : float = 3.5f
- chargeTime : float = 0.5f
- arrowForce : float = 0f
- arrowChargeMeter : Image
Start() : void
Update() : void
FixedUpdate() : void
SwordAttack
- animator : Animator
- attackDamage : float = 50f
- attackRange : float = 2f
- <readonly> hitMaskString : str
- playerStats : Stats.Stats
- isAttacking : bool
- canClick : bool
- <readonly> hitMaskNo : int
Start() : void
Update() : void
OnTriggerEnter(other:Collider) : void
CanAttackers() : void
CanAttackNo() : void
Arrow
- body : RigidBody
- attackDamage : float = 10f
- used(b) : bool = false
Start() : void
Update() : void
OnCollisionEnter(other:Collision) : void
AI
FSM
- stateStack : Stack<FSMState>
+ Update(gameObject:GameObject) : void
+ pushState(state:FSMState) : void
+ popState() : void
FSMState
+ Update(fsm:FSM, gameObject:GameObject) : void
CoopPlanner
+ planAgent(agent:GameObject, availableActions:HashSet<GoapAction>, worldState:HashSet<KeyValuePair<string, object>>, goal:HashSet<KeyValuePair<string, object>>) : Queue<GoapAction>
- buildGraph(parent:Node, leaves:List<Node>, usableActions:HashSet<GoapAction>, goal:HashSet<KeyValuePair<string, object>>) : bool
+ actionSubset(actions:HashSet<GoapAction>, removeMe:GoapAction) : HashSet<GoapAction>
# inState(test:HashSet<KeyValuePair<string, object>>, state:HashSet<KeyValuePair<string, object>>) : bool
# populateState(currentState:HashSet<KeyValuePair<string, object>>, stateChange:HashSet<KeyValuePair<string, object>>) : HashSet<KeyValuePair<string, object>>
Node
+ parent : Node
+ runningCost : float
+ state : HashSet<KeyValuePair<string, object>>
+ action : GoapAction
+ Node(parent:Node, runningCost:float, state:HashSet<KeyValuePair<string, object>>, action:GoapAction)
GoapAction
- preconditions : HashSet<KeyValuePair<string, object>>
- effects : HashSet<KeyValuePair<string, object>>
- inRange : bool = false
- radius : float = 1f
- cost : float = 1
- target : GameObject
+ PreconditionsGoapAction() : void
+ addPrecondition(key:string, value:object) : void
+ removePrecondition(key:string) : void
+ AddEffect(key:string, value:object) : void
+ RemoveEffect(key:string) : void
+ GoapAction() : void
+ reset() : void
+ isDone() : bool
+ checkProceduralPrecondition(agent:GameObject) : bool
+ perform(agent:GameObject) : bool
+ requiresInRange() : bool
+ isInRange() : bool
+ setInRange(inRange:bool) : void
+ addPrecondition(key:string, value:object) : void
+ removePrecondition(key:string) : void
+ AddEffect(key:string, value:object) : void
+ removeEffect(key:string) : void
IGoap
getWorldState() : HashSet<KeyValuePair<string, object>>
createGoalState() : HashSet<KeyValuePair<string, object>>
planFailed(failedGoal:HashSet<KeyValuePair<string, object>>) : void
planFound(goal:HashSet<KeyValuePair<string, object>>, actions:Queue<GoapAction>) : void
actionsFinished() : void
planAborted(aborter:GoapAction) : void
moveAgent(nextAction:GoapAction) : bool
_<nodird>_CoopAgent
- stateMachine : FSM
- idleState : FSM.FSMState
- moveToState : FSM.FSMState
- performActionState : FSM.FSMState
- availableActions : HashSet<GoapAction>
- currentActions : Queue<GoapAction>
- dataProvider : IGoap
- planner : CoopPlanner
Start() : void
Update() : void
+ addAction(a:GoapAction) : void
+ getAction(action:GoapAction) : GoapAction
+ removeAction(action:GoapAction) : void
# hasActionPlan() : bool
# actionsFinished() : void
# createMoveToState() : void
# createPerformActionState() : void
# changeState(s:FSM.FSMState) : void
# findDataProvider() : void
+ <readonly> getWorldState() : HashSet<KeyValuePair<string, object>> : string
+ <readonly> createGoalState() : HashSet<KeyValuePair<string, object>> : string
+ <readonly> planFailed(failedGoal:HashSet<KeyValuePair<string, object>>) : string
+ <readonly> planFound(goal:HashSet<KeyValuePair<string, object>>, actions:Queue<GoapAction>) : string
+ <readonly> actionsFinished() : string
+ <readonly> planAborted(aborter:GoapAction) : string
Creatures
Action_MeleeAttack
- hasAttacked : bool = false
- attackDamage : float = 10f
+ Action_MeleeAttack() : void
+ reset() : void
+ isDone() : bool
+ checkProceduralPrecondition(agent:GameObject) : bool
+ perform(agent:GameObject) : bool
+ requiresInRange() : bool
Action_EatFood
- foodIsEaten : bool = false
+ Action_EatFood()
+ reset() : void
+ isDone() : bool
+ checkProceduralPrecondition(agent:GameObject) : bool
+ perform(agent:GameObject) : bool
+ requiresInRange() : bool
Action_FindFood
- foodFound : bool = false
- callAFood : HashSet<string>
- food : TreeStrim
- sphereMask : LayerMask
- sphereRadius : float = 40
- viewAngle : float = 90
Action_FindFood()
Start() : void
reset() : void
isDone() : bool
checkProceduralPrecondition(agent:GameObject) : bool
perform(agent:GameObject) : bool
requiresInRange() : bool
BaseAICoop
- enableDebugging : bool = true
+ getWorldState() : HashSet<KeyValuePair<string, object>>
+ createGoalState() : HashSet<KeyValuePair<string, object>>
+ planFailed(failedGoal:HashSet<KeyValuePair<string, object>>) : void
+ planFound(goal:HashSet<KeyValuePair<string, object>>, actions:Queue<GoapAction>) : void
+ actionsFinished() : void
+ planAborted(aborter:GoapAction) : void
+ moveAgent(nextAction:GoapAction) : bool
+ StartMovingtoMoveBool, targetTransform) : void
Sloth
- stats : Stats.Stats
- stateMachine : FSM
- destinationSetter : AIDestinationSetter
- movementController : AIPath
- gizmoPoint : GizmoPoint
- sightDistance : float
- rigidBodyRadius : GameObject
- lifeLayerMask : LayerMask
Awake() : void
Start() : void
FixedUpdate() : void
+ <override> getWorldState() : HashSet<KeyValuePair<string, object>>
+ <override> createGoalState() : HashSet<KeyValuePair<string, object>>
+ <override> moveAgent(nextAction:GoapAction) : bool
+ <override> StartMovingtoMoveBool, targetTransform) : void
Sabertooth
- stats : Stats.Stats
- stateMachine : FSM
- destinationSetter : AIDestinationSetter
- movementController : AIPath
- <readonly> wanderRadius : int
Awake() : void
Start() : void
FixedUpdate() : void
+ <override> getWorldState() : HashSet<KeyValuePair<string, object>>
+ <override> createGoalState() : HashSet<KeyValuePair<string, object>>
+ <override> moveAgent(nextAction:GoapAction) : bool
+ <override> StartMovingtoMoveBool, targetTransform) : void
RandomlyRoam
- randomPositionObj : GameObject
- destinationSetter : AIDestinationSetter
- creature : BaseAICoop
- isRoamingAlready : bool
- isAvoidLakingBool : bool
- resetRandomPositionFrequency : float = 2f
- resetRandomPositionTimer : float = 0f
Awake() : void
Start() : void
Update() : void
Reset() : void
RagdollToggle
- animator : Animator
- boxCollider : BoxCollider
- allComponent : Behaviour
- childRigidBodies : List<Rigidbody>
- childHitBoxBodies : RigidBody[]
- currentState : bool
Awake() : void
Start() : void
ToggleRagdoll(isActive:bool) : void
FinishEating
- isInProgress : bool
- destinationSetter : AIDestinationSetter
- action_EatFood
- hasRun : bool
Start() : void
Update() : void
isInProgress() : bool
StopAnimation() : void
RandomlyCallFindFood
- callFrequency : float = 2f
- callTimer : float = 0f
- findFoodAction : Action_FindFood
Start() : void
Update() : void

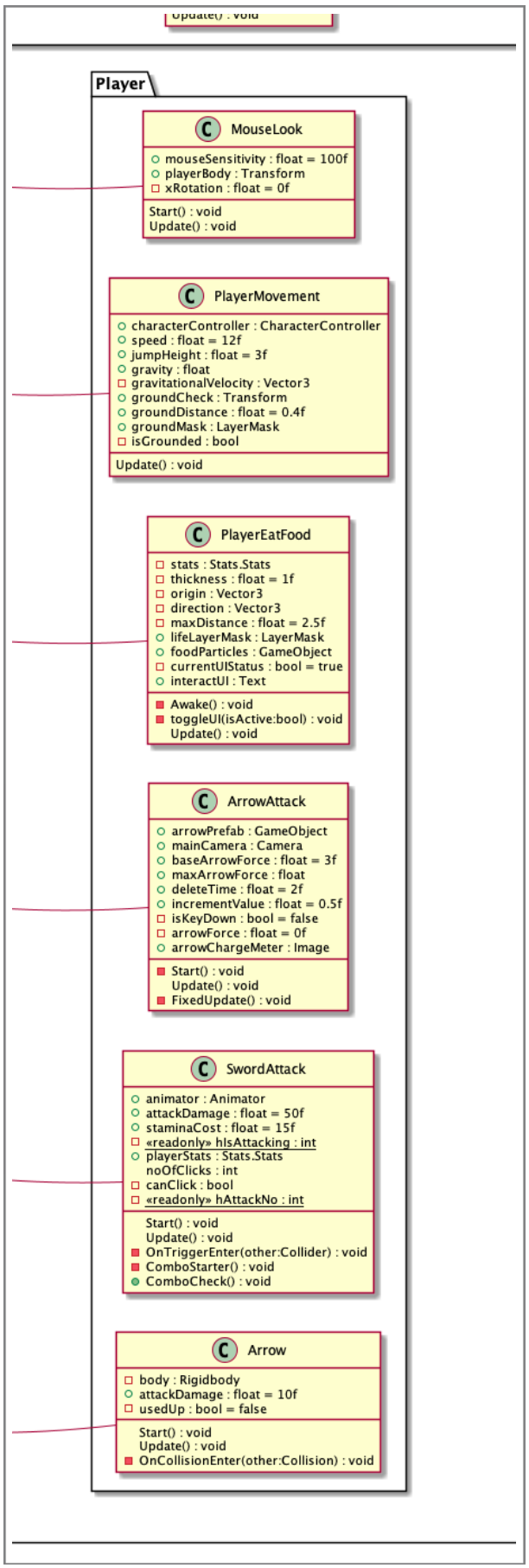
Managers


Managers
AStarMovement
targetPosition : Transform
seeker : Seeker
controller : CharacterController
path : Path
speed : float = 2
nextWaypointDistance : float = 3
currentWaypoint : int = 0
reachedEndOfPath : bool
Start() : void
OnPathComplete(p:Path) : void
Update() : void
FoodChain
«const» NAME_PLAYER : string = "player"
«const» NAME_SABERTOOTH : string = "sabertooth"
«const» NAME_SLOTH : string = "sloth"
«const» NAME_PLANT : string = "plant"
threats : Dictionary<String, HashSet<String>>
foods : Dictionary<String, HashSet<String>>
Awake() : void
addToChain(threat:String, food:String) : void
GetFood(name:String) : HashSet<String>
GetThreat(name:String) : HashSet<String>
MonoBehaviour
Stats

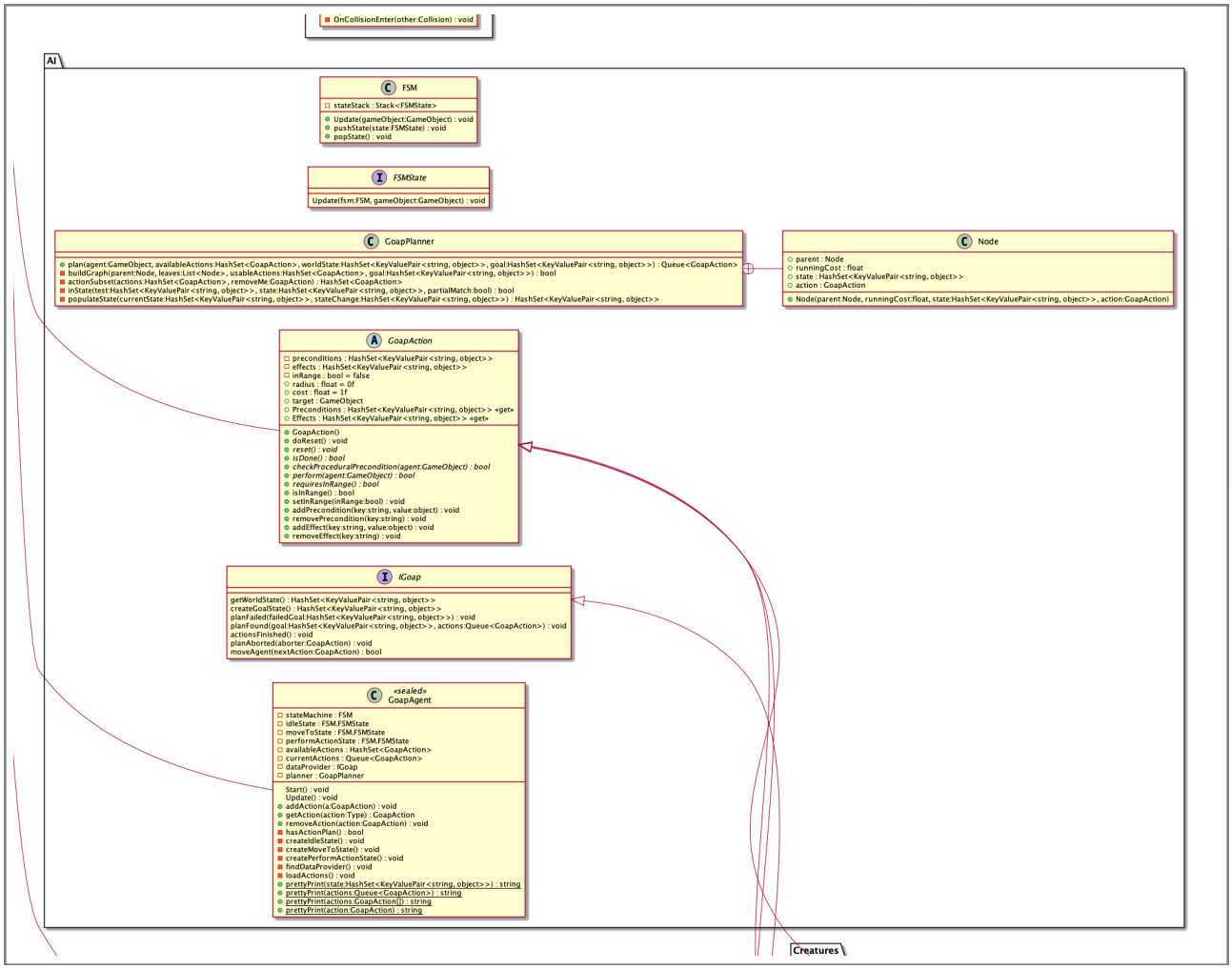
Stats


Stats
FoodStats
foodAmount : float = 10f
totalFoodAmount : float = 100f
isReadyToEat : bool = true
foodParticles : GameObject
Start() : void
isEaten() : bool
Update() : void
DepleteTotalFoodAmount(amount:float) : void
DepleteTotalFoodAmount() : void
DepleteTotalFoodAmount(centerPosition:Vector3, rotation:Quaternion) : void
SpawnFoodParticles(centerPosition:Vector3, rotation:Quaternion) : void
Food
targetFood : GameObject
Stats
health : float = 100
healthRegenRate : float = 2
healthThreshold : float = 50
hunger : float = 100
hungerDepletionRate : float = 2
hungerThreshold : float = 80f
stamina : float = 100
staminaRegenRate : float = 2
moveSpeed : float = 5
threatLevel : float = 0f
threatThresholds : float[]
bloodParticle : GameObject
ragdoll : RagdollToggle
isThereRagdoll : bool = false
isGoapAgentNotNull : bool
isAIPathNotNull : bool
Awake() : void
Update() : void
SetHealth(amount:float) : void
ApplyDamage(damage:float) : void
ApplyDamage(damage:float, centerPosition:Vector3, rotation:Quaternion) : void
DepleteHunger(decrement:float) : void
AddFoodAmount(amount:float) : void
IsHungry() : bool
GetHealth() : float
GetStamina() : float
ApplyStaminaReduction(staminaCost:float) : void
GetHunger() : float
GetHungerThreshold() : float
IncreaseThreatLevel(amount:float) : void
GetThreatLevel() : float
IsThreatPastThreshold() : int
IsDead() : bool
GetProperSpeed() : float
Death() : void
SpawnBloodParticles(centerPosition:Vector3, rotation:Quaternion) : void
StatsUIController
healthBar : Image
staminaBar : Image
hungerBar : Image
hungerThresholdBar : Image
creatureStats : global::Stats.Stats
_enabled : bool = true
Awake() : void
Update() : void
FixedUpdate() : void
PlayerStatsUi
healthBar : Image
staminaBar : Image
hungerBar : Image
playerStats : Stats.Stats
Start() : void
Update() : void
Player

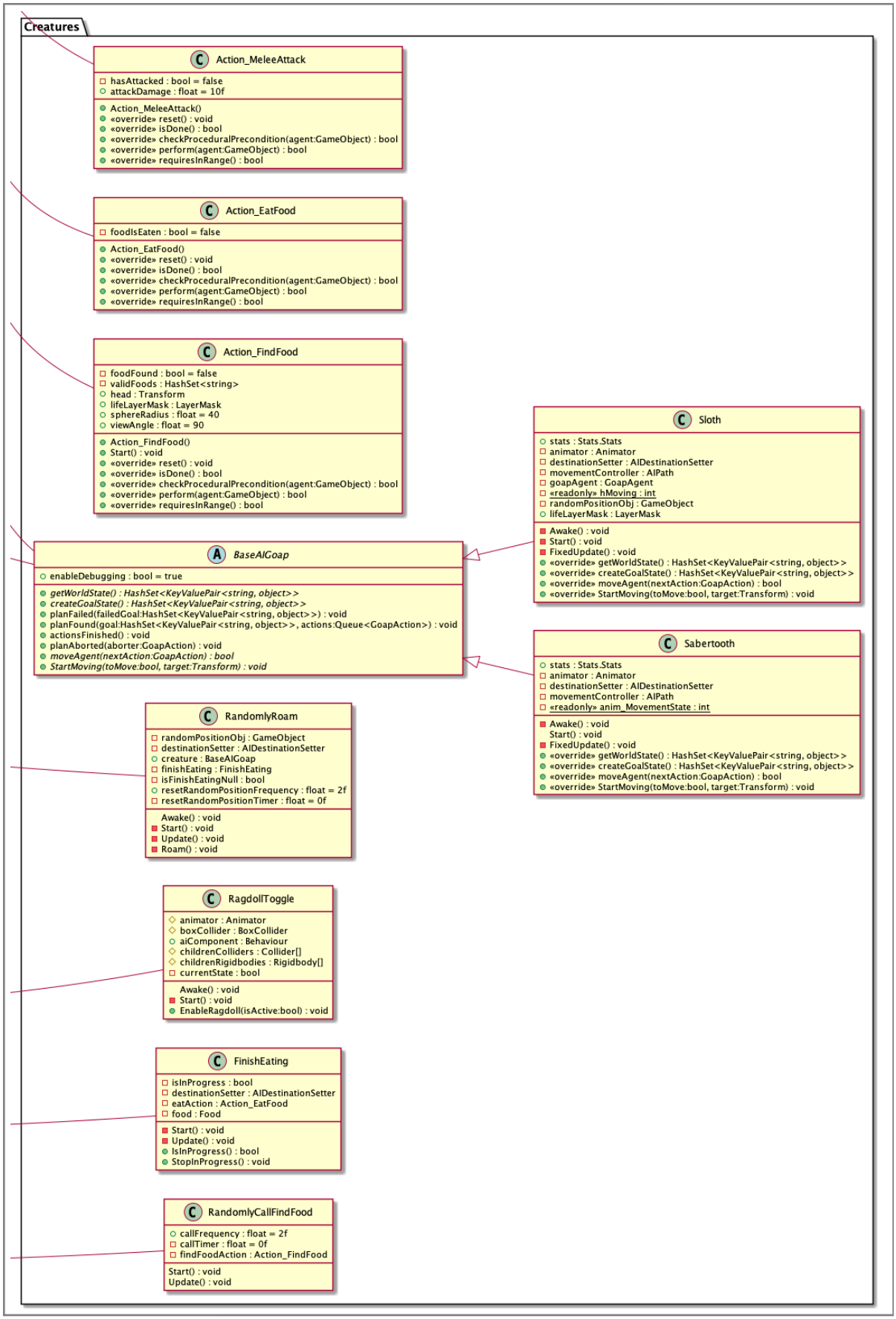
Player


Update() : void
Player
MouseLook
mouseSensitivity : float = 100f
playerBody : Transform
xRotation : float = 0f
Start() : void
Update() : void
PlayerMovement
characterController : CharacterController
speed : float = 12f
jumpHeight : float = 3f
gravity : float
gravitationalVelocity : Vector3
groundCheck : Transform
groundDistance : float = 0.4f
groundMask : LayerMask
isGrounded : bool
Update() : void
PlayerEatFood
stats : Stats.Stats
thickness : float = 1f
origin : Vector3
direction : Vector3
maxDistance : float = 2.5f
lifeLayerMask : LayerMask
foodParticles : GameObject
currentUIStatus : bool = true
interactUI : Text
Awake() : void
toggleUI(isActive:bool) : void
Update() : void
ArrowAttack
arrowPrefab : GameObject
mainCamera : Camera
baseArrowForce : float = 3f
maxArrowForce : float
deleteTime : float = 2f
incrementValue : float = 0.5f
isKeyDown : bool = false
arrowForce : float = 0f
arrowChargeMeter : Image
Start() : void
Update() : void
FixedUpdate() : void
SwordAttack
animator : Animator
attackDamage : float = 50f
staminaCost : float = 15f
«readonly» hIsAttacking : int
playerStats : Stats.Stats
noOfClicks : int
canClick : bool
«readonly» hAttackNo : int
Start() : void
Update() : void
OnTriggerEnter(other:Collider) : void
ComboStarter() : void
ComboCheck() : void
Arrow
body : Rigidbody
attackDamage : float = 10f
usedUp : bool = false
Start() : void
Update() : void
OnCollisionEnter(other:Collision) : void

AI


OnCollisionEnter(other:Collision) : void
AI
FSM
stateStack : Stack<FSMState>
Update(gameObject:GameObject) : void
pushState(state:FSMState) : void
popState() : void
FSMState
Update(fsm:FSM, gameObject:GameObject) : void
GoapPlanner
plan(agent:GameObject, availableActions:HashSet<GoapAction>, worldState:HashSet<KeyValuePair<string, object>>, goal:HashSet<KeyValuePair<string, object>>) : Queue<GoapAction>
buildGraph(parent:Node, leaves:List<Node>, usableActions:HashSet<GoapAction>, goal:HashSet<KeyValuePair<string, object>>) : bool
actionSubset(actions:HashSet<GoapAction>, removeMe:GoapAction) : HashSet<GoapAction>
inState(test:HashSet<KeyValuePair<string, object>>, state:HashSet<KeyValuePair<string, object>>, partialMatch:bool) : bool
populateState(currentState:HashSet<KeyValuePair<string, object>>, stateChange:HashSet<KeyValuePair<string, object>>) : HashSet<KeyValuePair<string, object>>
Node
parent : Node
runningCost : float
state : HashSet<KeyValuePair<string, object>>
action : GoapAction
Node(parent:Node, runningCost:float, state:HashSet<KeyValuePair<string, object>>, action:GoapAction)
GoapAction
preconditions : HashSet<KeyValuePair<string, object>>
effects : HashSet<KeyValuePair<string, object>>
inRange : bool = false
radius : float = 0f
cost : float = 1f
target : GameObject
Preconditions : HashSet<KeyValuePair<string, object>> «get»
Effects : HashSet<KeyValuePair<string, object>> «get»
GoapAction()
doReset() : void
reset() : void
isDone() : bool
checkProceduralPrecondition(agent:GameObject) : bool
perform(agent:GameObject) : bool
requiresInRange() : bool
isInRange() : bool
setInRange(inRange:bool) : void
addPrecondition(key:string, value:object) : void
removePrecondition(key:string) : void
addEffect(key:string, value:object) : void
removeEffect(key:string) : void
IGoap
getWorldState() : HashSet<KeyValuePair<string, object>>
createGoalState() : HashSet<KeyValuePair<string, object>>
planFailed(failedGoal:HashSet<KeyValuePair<string, object>>) : void
planFound(goal:HashSet<KeyValuePair<string, object>>, actions:Queue<GoapAction>) : void
actionsFinished() : void
planAborted(aborter:GoapAction) : void
moveAgent(nextAction:GoapAction) : bool
«sealed»
GoapAgent
stateMachine : FSM
idleState : FSM.FSMState
moveToState : FSM.FSMState
performActionState : FSM.FSMState
availableActions : HashSet<GoapAction>
currentActions : Queue<GoapAction>
dataProvider : IGoap
planner : GoapPlanner
Start() : void
Update() : void
addAction(a:GoapAction) : void
getAction(action:Type) : GoapAction
removeAction(action:GoapAction) : void
hasActionPlan() : bool
createIdleState() : void
createMoveToState() : void
createPerformActionState() : void
findDataProvider() : void
loadActions() : void
prettyPrint(state:HashSet<KeyValuePair<string, object>>) : string
prettyPrint(actions:Queue<GoapAction>) : string
prettyPrint(actions:GoapAction[]) : string
prettyPrint(action:GoapAction) : string
Creatures

Creatures


Creatures
Action_MeleeAttack
hasAttacked : bool = false
attackDamage : float = 10f
Action_MeleeAttack()
«override» reset() : void
«override» isDone() : bool
«override» checkProceduralPrecondition(agent:GameObject) : bool
«override» perform(agent:GameObject) : bool
«override» requiresInRange() : bool
Action_EatFood
foodIsEaten : bool = false
Action_EatFood()
«override» reset() : void
«override» isDone() : bool
«override» checkProceduralPrecondition(agent:GameObject) : bool
«override» perform(agent:GameObject) : bool
«override» requiresInRange() : bool
Action_FindFood
foodFound : bool = false
validFoods : HashSet<string>
head : Transform
lifeLayerMask : LayerMask
sphereRadius : float = 40
viewAngle : float = 90
Action_FindFood()
Start() : void
«override» reset() : void
«override» isDone() : bool
«override» checkProceduralPrecondition(agent:GameObject) : bool
«override» perform(agent:GameObject) : bool
«override» requiresInRange() : bool
Sloth
stats : Stats.Stats
animator : Animator
destinationSetter : AIDestinationSetter
movementController : AIPath
goapAgent : GoapAgent
«readonly» hMoving : int
randomPositionObj : GameObject
lifeLayerMask : LayerMask
Awake() : void
Start() : void
FixedUpdate() : void
«override» getWorldState() : HashSet<KeyValuePair<string, object>>
«override» createGoalState() : HashSet<KeyValuePair<string, object>>
«override» moveAgent(nextAction:GoapAction) : bool
«override» StartMoving(toMove:bool, target:Transform) : void
BaseAIGoap
enableDebugging : bool = true
getWorldState() : HashSet<KeyValuePair<string, object>>
createGoalState() : HashSet<KeyValuePair<string, object>>
planFailed(failedGoal:HashSet<KeyValuePair<string, object>>) : void
planFound(goal:HashSet<KeyValuePair<string, object>>, actions:Queue<GoapAction>) : void
actionsFinished() : void
planAborted(aborter:GoapAction) : void
moveAgent(nextAction:GoapAction) : bool
StartMoving(toMove:bool, target:Transform) : void
Sabertooth
stats : Stats.Stats
animator : Animator
destinationSetter : AIDestinationSetter
movementController : AIPath
«readonly» anim_MovementState : int
Awake() : void
Start() : void
FixedUpdate() : void
«override» getWorldState() : HashSet<KeyValuePair<string, object>>
«override» createGoalState() : HashSet<KeyValuePair<string, object>>
«override» moveAgent(nextAction:GoapAction) : bool
«override» StartMoving(toMove:bool, target:Transform) : void
RandomlyRoam
randomPositionObj : GameObject
destinationSetter : AIDestinationSetter
creature : BaseAIGoap
finishEating : FinishEating
isFinishEatingNull : bool
resetRandomPositionFrequency : float = 2f
resetRandomPositionTimer : float = 0f
Awake() : void
Start() : void
Update() : void
Roam() : void
RagdollToggle
animator : Animator
boxCollider : BoxCollider
aiComponent : Behaviour
childrenColliders : Collider[]
childrenRigidbodies : Rigidbody[]
currentState : bool
Awake() : void
Start() : void
EnableRagdoll(isActive:bool) : void
FinishEating
isInProgress : bool
destinationSetter : AIDestinationSetter
eatAction : Action_EatFood
food : Food
Start() : void
Update() : void
IsInProgress() : bool
StopInProgress() : void
RandomlyCallFindFood
callFrequency : float = 2f
callTimer : float = 0f
findFoodAction : Action_FindFood
Start() : void
Update() : void