BlackDeath

-speedDowngrade:float

+BlackDeath()
-Start()
#Symptoms(Actor _actor):IEnumerator<<override>>

VirusS

#waitTime:float
#freezeTime:float
#freezeProb:int
#isFrozen:bool

+VirusS()
-Start()
#Symptoms(Actor _actor):IEnumerator<<override>>

VirusA

#downgradePercentage:float
#nerfSpeed:float

+VirusA()
-Start()
Symptoms(Actor _actor):IEnumerator<<override>>

Disease

+symptomsHandler:SymptomsHandler
#onsetTime:float
#timeUntilDeath:float
-mShader:Shader
#type:DiseaseType
+OnsetTime:float
+TimeUntilDeath:float
+MShader:Shader
+Type:DiseaseType

+Disease()
#Symptoms(Actor _actor):IEnumerator<>

Survivor

-vaccines:List<Transform>
-mColliders:Colliders
-vaccinePosition:Vector3
-offset:Vector3
-scanTime:float
-mRigid:RigidBody

#Start ():<<override>>
#OnCollisionEnter(Collision collision)
#Infected(Disease dis):IEnumerator<<override>>
-SearchCure():IEnumerator
-OnTriggerEnter(Collider other)

Denizen

-currentPoint
#pauseMovement:bool

#Start():<<override>>
Movement():IEnumerator<<override>>

Vaccine

-diseaseToCure:DiseaseType
-immunity:float
+DiseaseToCure:DiseaseType

-OnCollisionEnter(collision:Collision)

<<enum>> DiseaseType

VirusA,
VirusS,
BlackDeath

Grudger

-distance:float
-distanceToDetect:float
-currentPoint:int
-timeToLeave:float

#Start():<<override>
#Movement():IEnumerator<<override>>
+FollowPlayer():IEnumerator<<virtual>>

AI

-pointsContainer:Transform
-pointsL:Transform[ ]
-player:Transform
-navmesh:NavMeshAgent

#Start():<<override>>
#UpdateStats(float _speed):<<override>>
#Movement():IEnumerator <<override>>
#Death:IEnumerator <<override>>

DrawPoints

-points:Transform[ ]

+OnDrawGizmos()

GameController

-door:GameObject

+Start()
-GameOver()
-EpicWin()

PlayerController

-player:Transform
-cam:Transform
-pivot:Transform

-Update()

Player

+OnGameOver:GameHandler<<event>>
+OnEpicWin:GameHandler<<event>>
-hasTheKey:bool
-camPivot:Transform
-rotSpeed:float

#Start():<<override>>
#Death():IEnumerator<<override>>
#UpdateStats(float _speed)
#Movement():IEnumerator<<override>>
#OnCollisionEnter(Collision collision):<<override>>

Actor

+healingHandler:Action<Disease, float>
+statsHandler:Action<float>
-mRenderer:Renderer
-mShader:Shader
#mAnimator:Animator
#speed:float
-initialSpeed:float
#diseaseList:List<Disease>
-Disease[] initialDiseases;
+DiseaseList:List<Disease>
+IsHealing:bool
+IsDead:bool
-virusAProb:float
-virusSProb:float
-blackDeathProb:float
#virusAResistance:float #virusSResistance:float
#blackDeathResistance:float
#virusATimeInfected:float #virusSTimeInfected:float
#blackdeathTimeInfected:float
#Speed:float
+InitialSpeed:float

#Movement():<<abstracct>>
#Death():<>
#UpdateStats(float _speed):<>
#movementCoroutine():Coroutine
#Start():<<Virtual>>
-VaccineEffects(Disease disease, float reduceProb)
#OnCollisionEnter(Collision collision):<<Virtual>>
#Infected(Disease dis):IEnumerator <<Virtual>>
-Probability(DiseaseType disease):bool