ABASECHARACTER:

Die()

{

Turn off ticking

ServerDropItem

If(HasAuthority)

NetMulticastOnDeath

}

NetMulticastOnDeath\_Implementation

{

Turn off capsule & Sphere collision & visibility

bIsDead = true

Turn on Mesh collision for ragdoll

}

PerformRayCast(FName, FHitResult)

{

LineTraceSingleByProfile from the camera’s position to InteractRange in its direction

Return the Boolean result of the cast

}

ServerDropItem

{

If(HeldObject is not nullptr)

{

HeldObject->Drop

Set HeldObject to nullptr

Set bIsInteracting to false

}

}

Interact

{

If we’re holding an object

ServerDropItem

Else

{

PerformRayCast against GrabbableTrace

If that succeeded

Grab what we hit

}

}

Grab

{

Safety check that we’re not holding anything and that what we grabbed is grabbable

If HasAuthority

{

HeldObject = what we hit

PickupObject

}

}

ServerThrowObject

{

If HasAuthority

ThrowObject

}

ThrowObject

{

Call Throw on the held object

Set HeldObject to nullptr and bIsInteracting to false

}

APLAYERCHARACTER

BeginPlay

{

Start a timer to call PostBeginPlay after a second

}

AssignTeams

{

Check the gamestate

For each possible player in gamestate if the lowest one isn’t logged in sign in as them

And set that as our team

Otherwise keep going to higher numbers until we’re out

}

Multicast\_AssignDefaultMaterial

{

Set our DefaultMaterial to the corresponding team’s material in gamestate

}

PostBeginPlay

{

Call NetMulticastSetColor which actually assigns our material to mesh

Wait so that the game has time to properly load the things we need for this

}

Client\_BeingChased

{

If We’re chased

Fade our normal music and fade in chase music

Else

The opposite

}

Die

{

Set this transform as our respawnlocation so we can hopefully respawn here

Call super’s die

Reset the playerstate

Get our controller and disable input to it

Hide our capsule

Start a timer to call Respawn

}

Respawn

{

If we have a controller and authority

Call RespawnPlayer in gamemode

Stop both musics

Destroy ourselves

}

Interact

{

Call super’s interact

If after we still aren’t interacting

Perform a raycast against grabbabletrace

If that succeeded and what we hit has the loot tag

Play the loot sound

Tell our inventory to ServerCollectLoot

}

PlaceTrap

{

If we have any traps and our raycast succeeds in hitting something

Tell the inventory to PlaceTrap

}

GUARDCHARACTER

BeginPlay

{

If we have authority

Setup our onoverlap

Start a timer to call postbegin play

Set our gamemode variable

}

OnPawnSee

{

Return if the target is already dead

If we already have a target

Switch to this target if they’re closer than our current

If they aren’t, return

Otherwise set our target to this new one

SetGuardState to attack

}

ResetPatrol

{

If we have a target tell it we lost it so it can change its music back

SetGuardState to patrol and move to a new patrol point

}

DetermineGuardState

{

If we have a target and can see them attack

If we have a target and can’t see them search

If we have no target patrol

}

HandleAI

{

Switch on our state

If its patrol

If we have no patrol point move to the next

If we have a patrol point and it is in our room

If we’re really close to it consider it done and get a new one

Else get a new one

If its search

If we’re where the player last was and can’t find them check their last door

If we’re not at the last seen place keep moving

If the guard’s search timer is over, give up and go back to patrol

If its Attack

Walk directly at the target until we hopefully touch

}

MoveToNextPatrolPoint

{

If we have authority

FindNewPatrolPoint

Simplemoveto this point

}

FindNewPatrolPoint

{

Get the door array for the room we’re in

Choose a random door in the room as the next target, excluding the door we came from

}

isPatrolPointInRoom

{

Get this room’s door actors

Check if our target is in that array, if it is return true, else false

}

UHEALTHCOMPONENT

InitializeComponent

{

Setup values

Set it so our owner bCanBeDamaged = true

Subscribe with our OnTakeAnyDamage function to our owner’s event

}

OnTakeAnyDamage

{

If we’re already dead return

Subtract the damage from our currenthealth

Clamp our current health between 0 and max

If currenthealth <= 0

OnDeath.Broadcast

}

ReplenishHealth

{

Heal the amount passed in up to maximum

}

UINVENTORYCOMPONENT

DidFindTrap

{

Get a random number between 1 & 100

If that number is <= the loot’s ChanceToFindTrap

Return true

Else

Return false

}

AddRandomTrap

{

If our owner HasAuthority

Cycle through our inventory and add a random trap to the first open slot we find

}

Server\_CollectLoot

{

If owner hasauthoity

CollectLoot

}

CollectLoot

{

If we DidFindTrap & HasOpenSlot

AddRandomTrap

Tell the loot to die (multicast)

Add a random score to the player

}

SpawnTrap

{

If the selected inventory slot actually has a trap in it and our trapcount is > 0

Setup the spawnparams and spawn a trap of that slot’s type

Set the trap’s owner to ours

Remove that trap from our inventory, decrement trapcount

}