FIT2097 ASSIGNMENT 3

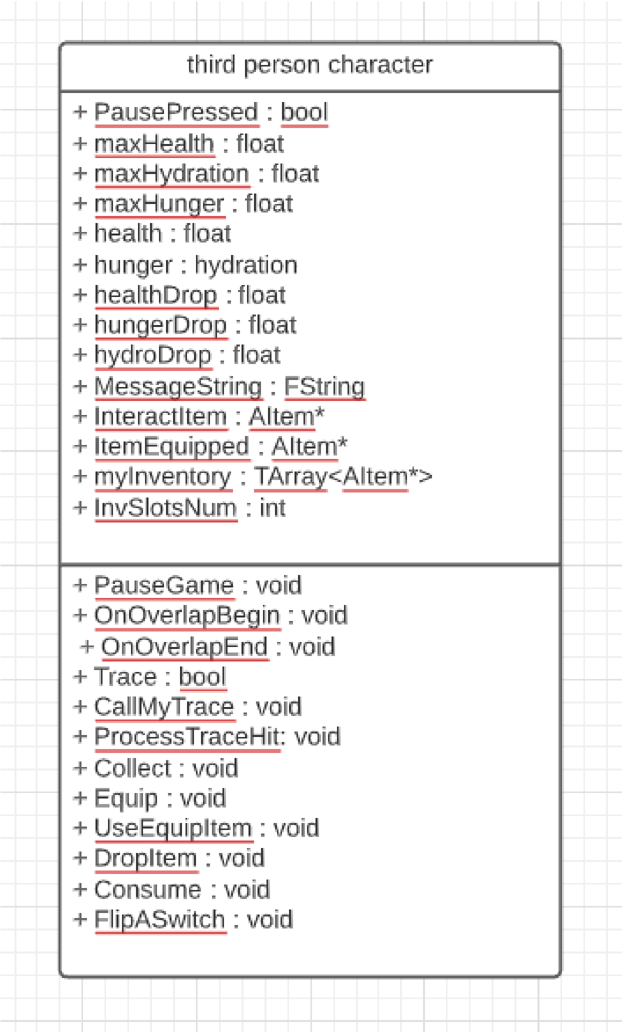
SURVIVAL GAME PROTOTYPE

NAME: ZHI WEI LOH ID: 31123074

PROGRAM DESIGN

CHARACTER CLASS

Third person character use in game.



third person character

+ PausePressed : bool
+ maxHealth : float
+ maxHydration : float
+ maxHunger : float
+ health : float
+ hunger : hydration
+ healthDrop : float
+ hungerDrop : float
+ hydroDrop : float
+ MessageString : FString
+ InteractItem : AItem*
+ ItemEquipped : AItem*
+ myInventory : TArray<AItem*>
+ InvSlotsNum : int

+ PauseGame : void
+ OnOverlapBegin : void
+ OnOverlapEnd : void
+ Trace : bool
+ CallMyTrace : void
+ ProcessTraceHit: void
+ Collect : void
+ Equip : void
+ UseEquipItem : void
+ DropItem : void
+ Consume : void
+ FlipASwitch : void

variables are mostly simple as how it is name,

for InteractItem , it is the item player’s raycast hits, InvSlotNum is used to check that item held by player do not exist inventory slots available in HUD.

Functions:

- pauseGame : bound with keypress ‘P’ which change the PausePressed variable read by gamemode to change game state.

- OnOverlapBegin : bound with the collision capsule , which is mostly use to detect end point and pit traps which broadcast gameover event.

- Trace : perform line tracing and return Boolean on whether actor is hit

- CallMyTrace: sets parameters to be use for raycast and call the trace function

- ProcessTraceHit : Handle related implementation base on trace result, in this case, it shows the information of item if hit or returns status of switch if hit.

- Tick: Player has constant hunger and hydration drop and when both reach zero health starts to drop and check for when health equals or lesser then zero game over event is broadcasted.

- Collect: bound to keypress ‘C’ to collect item player is interacting(raycasting) with to inventory.

- Equip: bound to keypress ’E’ and button press in inventory HUD to equip item selected as player tool which can be use or drop.

- UseEquipItem : use the item equipped by the player by calling the useitem function of the item and remove it from the inventory. Also displays related message to player.

- DropItem : drop equipped item nearby and remove it from inventory.

- Consume : bound to keypress ‘R’ , specifically used for consumable pickup items when interacted to consume directly.

- FlipASwitch : bound to keypress ‘F’ for flipping switch when nearby to switch which broadcast flipswitch events.

All events will be discussed in separate section.

FINAL ADD ONS

myPet variable to point to pet.

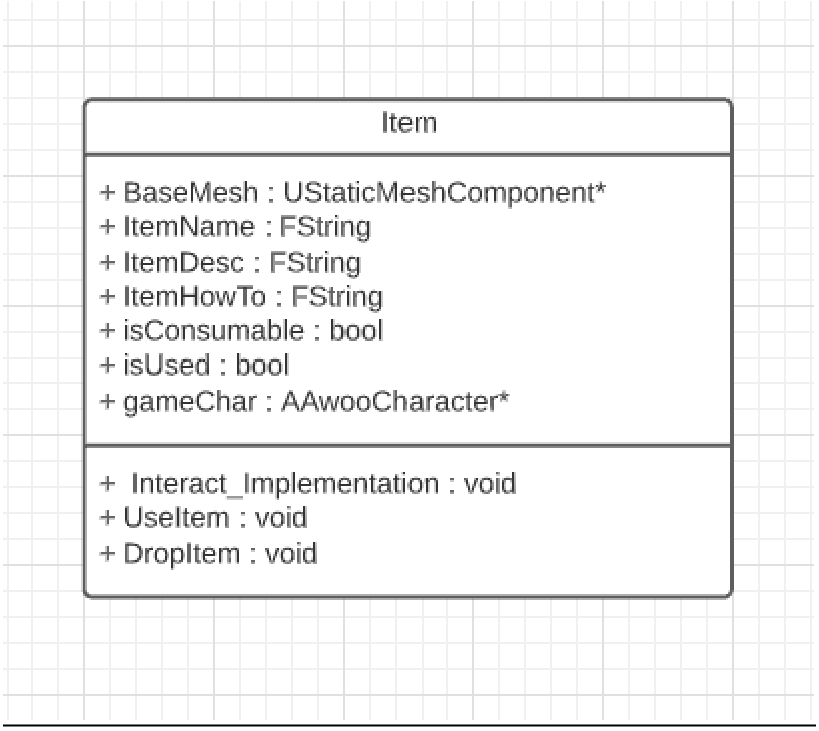
isProtected Boolean, isContacted Boolean used for altering values of damage received.

ReceiveDamage function to generalize changing values of health and applying post process.

ReleaseSkill function used to order pet and apply effects from skills.

ITEM CLASS (UE4 ACTOR CLASS, INTERACTABLE INTERFACE)

Interactable items in game.



Item

+ BaseMesh : UStaticMeshComponent*
+ ItemName : FString
+ ItemDesc : FString
+ ItemHowTo : FString
+ isConsumable : bool
+ isUsed : bool
+ gameChar : AAwooCharacter*

+ Interact_Implementation : void
+ UseItem : void
+ DropItem : void

isConsumable checks whether the item can be consume in other words, whether it is a pickup.

isUsed is used to check whether the player reach use condition so that the item can be reused by player again without being removed from inventory after use directly.

gameChar is a reference to game character.

Functions:

Interact_Implementation : called by character collect function so that the item will be hidden in game with its collision disabled.

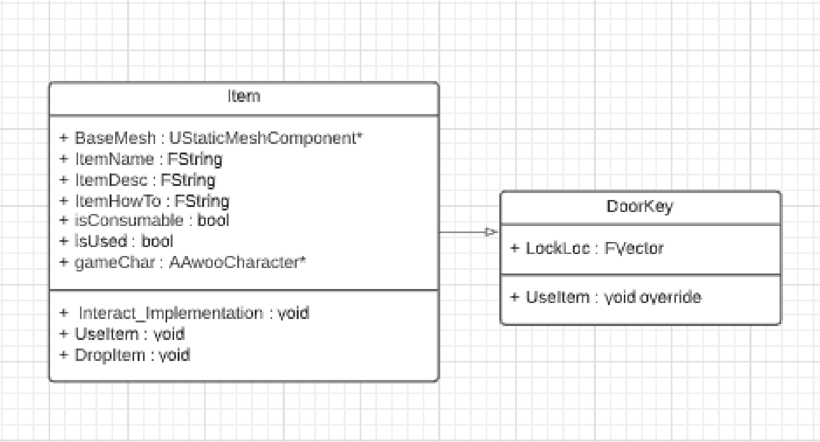
UseItem : handles item functionality, virtual function to be override by children class.

DropItem : drop item at location near to player and make item visible and enable collision again.

DOORKEY CLASS (CHILD CLASS OF ITEM CLASS)

Key for unlocking locked switches and doors.

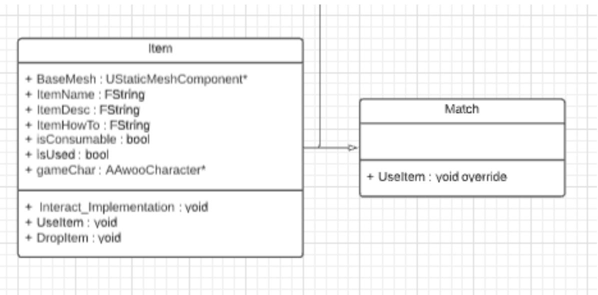
(only additional properties differ from parent class will be mentioned)



LockLoc : Used to check whether key is near to its connected switch or door so that the key can only be used when near to it.

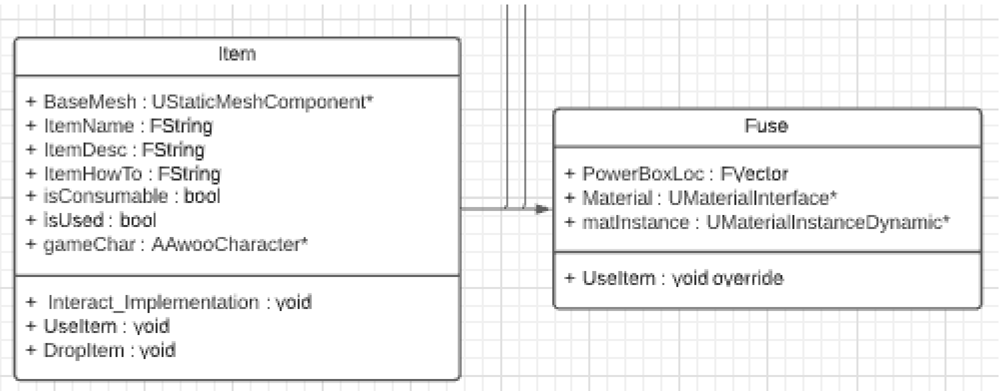
UseItem() : check distance between character and the lock, if true then broadcast unlock event and set isUsed to true, if not display message to player on distance too far.

MATCH CLASS(ITEM CHILD CLASS)



An item with light up event that broadcast function bound in torchlight class to light it up when this item is used by player.

FUSE CLASS (ITEM CHILD CLASS)


Item
+ BaseMesh : UStaticMeshComponent*
+ ItemName : FString
+ ItemDesc : FString
+ ItemHowTo : FString
+ isConsumable : bool
+ isUsed : bool
+ gameChar : AAwooCharacter*
+ Interact_Implementation : void
+ UseItem : void
+ DropItem : void
Fuse
+ PowerBoxLoc : FVector
+ Material : UMaterialInterface*
+ matInstance : UMaterialInstanceDynamic*
+ UseItem : void override

PowerBoxLoc : used to check whether fuse is only use around corresponding powerbox

PICKUPITEM CLASS (CHILD CLASS OF ITEM CLASS)

All pickups in game which can be used for altering player stats.

Variables:

- boostValue (float)
- boostType (EPickupType enum)

boostType can be PT_HEALTH , PT_FOOD, PT_WATER of the player while boostValue is the amount to increase or decrease the specified type.

Override function :

UseItem() : check for gameChar reference and boostType then add on boostValue to the boostType of the player.

PETFOOD CLASS (CHILD CLASS OF ITEM CLASS)

Food item used to feed pet.

Override function :

UseItem() : broadcast feed event when used by player and trigger pet updates, is unable to use by player when pet is not owned.

STUNGRENADE CLASS (CHILD CLASS OF ITEM CLASS)

Variables:

stunSec , stunRadius, offsetDistance : used to set the grenade

ParticleFX and effect for Niagara particles

startSecond, currentSecond used to track grenade time.

Boolean values , stunStart and finishAim used for placing and triggering grenade.

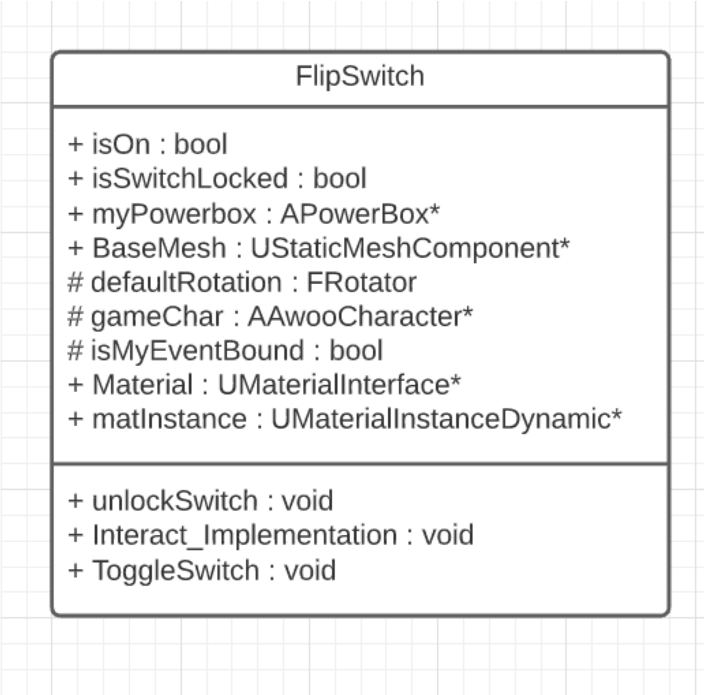
Override function :

UseItem() : When called for the first time draw debug helper for player to locate stun location,

When player call useitem again the stun grenade will then be set off.

FLIPSWITCH CLASS (UE4 ACTOR CLASS, INTERACTABLE INTERFACE)

Switch which can be toggle on and off by player key press when nearby.



FlipSwitch

+ isOn : bool
+ isSwitchLocked : bool
+ myPowerbox : APowerBox*
+ BaseMesh : UStaticMeshComponent*
# defaultRotation : FRotator
# gameChar : AAwooCharacter*
# isMyEventBound : bool
+ Material : UMaterialInterface*
+ matInstance : UMaterialInstanceDynamic*

+ unlockSwitch : void
+ Interact_Implementation : void
+ ToggleSwitch : void

isOn to be access by pit trap bridge to activate itself and deactivate accordingly.

gameChar reference to check whether player is near enough to flip the switch.

isMyEventBound used to ensure binding only happens once when game character reference is established.

defaultRotation used to set rotation so player can see further changes then messages when switch is activated.

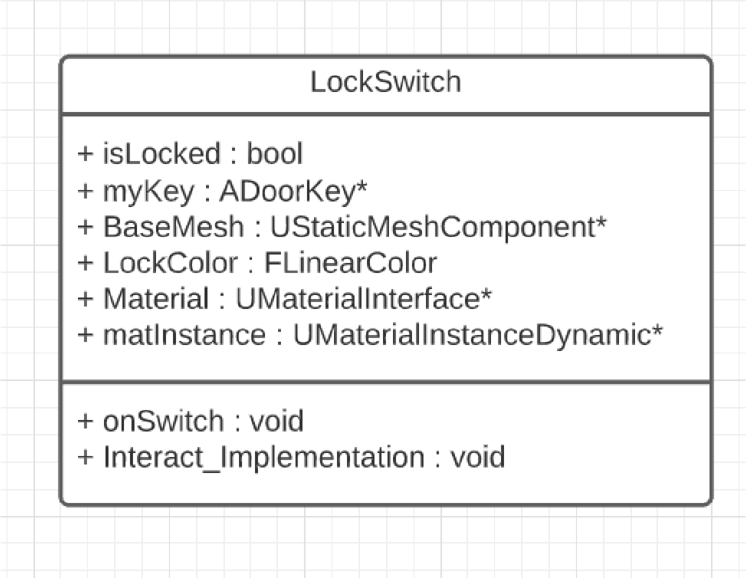
isSwitchLocked is used to lock the switch when it is attached to powerbox and is unlock when powered.

Function:

ToggleSwitch : change bool isOn if player is near enough which is bound to flipswitch event of player.

Interact_Implementation : show player whether the lock is on or off in form of display text when interacted.

LOCKEDSWITCH CLASS (UE4 ACTOR CLASS, INTERACTABLE INTERFACE)



LockSwitch

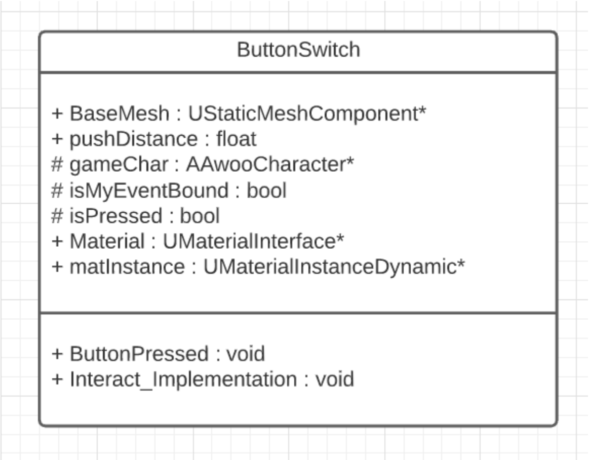
+ isLocked : bool
+ myKey : ADoorKey*
+ BaseMesh : UStaticMeshComponent*
+ LockColor : FLinearColor
+ Material : UMaterialInterface*
+ matInstance : UMaterialInstanceDynamic*

+ onSwitch : void
+ Interact_Implementation : void

It has almost similar variables and functions to FlipSwitch class except a custom variable myKey which points to aDoorKey item class so that the switch can only be unlock by specific key.

There is also a similar class LockedDoor which is made when misunderstanding the assignment brief so that is acts the same as locked switch just that it doesn't connects to a bridge.

BUTTONSWITCH (UE4 ACTOR CLASS, INTERACTABLE INTERFACE)



ButtonSwitch

+ BaseMesh : UStaticMeshComponent*
+ pushDistance : float
# gameChar : AAwooCharacter*
# isMyEventBound : bool
# isPressed : bool
+ Material : UMaterialInterface*
+ matInstance : UMaterialInstanceDynamic*

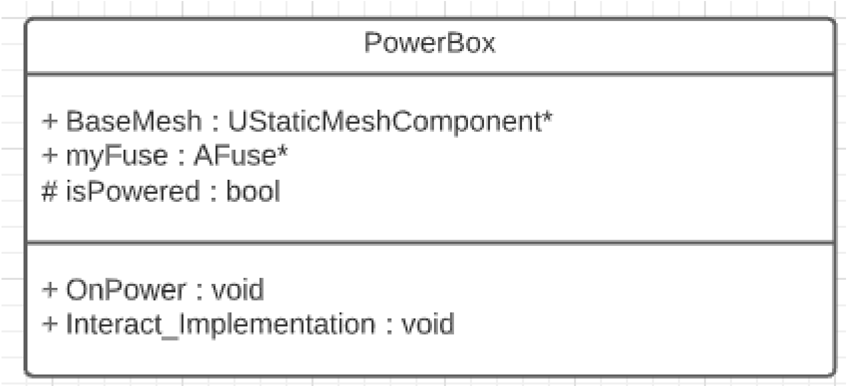
+ ButtonPressed : void
+ Interact_Implementation : void

Used to toggle lightpad.

pushDistance : maximum distance from player that button can be pushed.

DProduct: newly added to ensure interaction is only available when player stands in front.

POWERBOX (UE4 ACTOR CLASS, INTERACT INTERFACE)



PowerBox

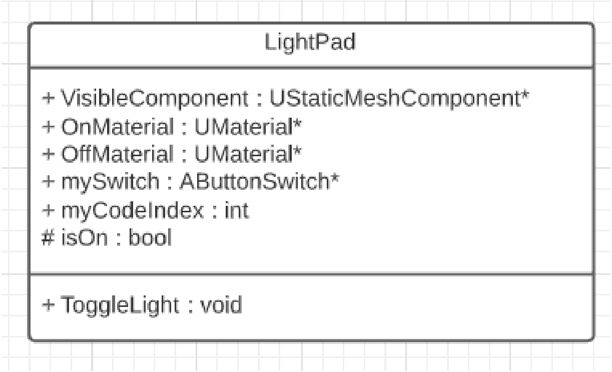
+ BaseMesh : UStaticMeshComponent*
+ myFuse : AFuse*
# isPowered : bool

+ OnPower : void
+ Interact_Implementation : void

myFuse : used to power it to unlock the switch on it.

OnPower : toggle isPowered.

LIGHTPAD (UE4 ACTOR CLASS)



LightPad

+ VisibleComponent : UStaticMeshComponent*
+ OnMaterial : UMaterial*
+ OffMaterial : UMaterial*
+ mySwitch : AButtonSwitch*
+ myCodeIndex : int
# isOn : bool

+ ToggleLight : void

Changes between OnMaterial and OffMAterial based on isOn.

myCodeIndex is used to tell the bridge which digit in binary code does it belong.

ENDPLATFORM CLASS ,MYTRAP CLASS(UE4 ACTOR CLASS)

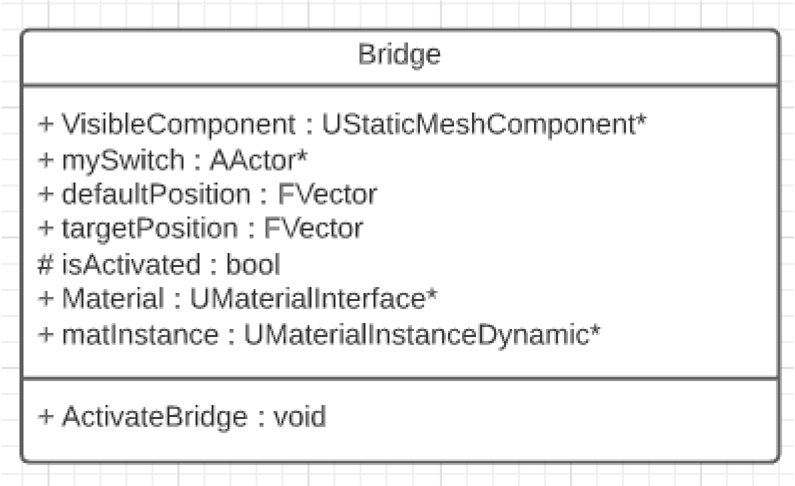
Endplatform class is a simple basic mesh which rotates stationarily and has the same collision channel with the player character “Trigger”, thus the on overlap function of the player will check when player overlap with this class and broadcast events accordingly.

MyTrap works the same as well by having the same collision channel and being checked by player overlap function.

TORCHLIGHT CLASS (UE4 ACTOR CLASS, INTERACT INTERFACE)

A mesh with point light that is toggle on when match is used by player nearby. The interact_implementation function also hints player to light it up when interacted.

BRIDGE CLASS (UE4 ACTOR CLASS)



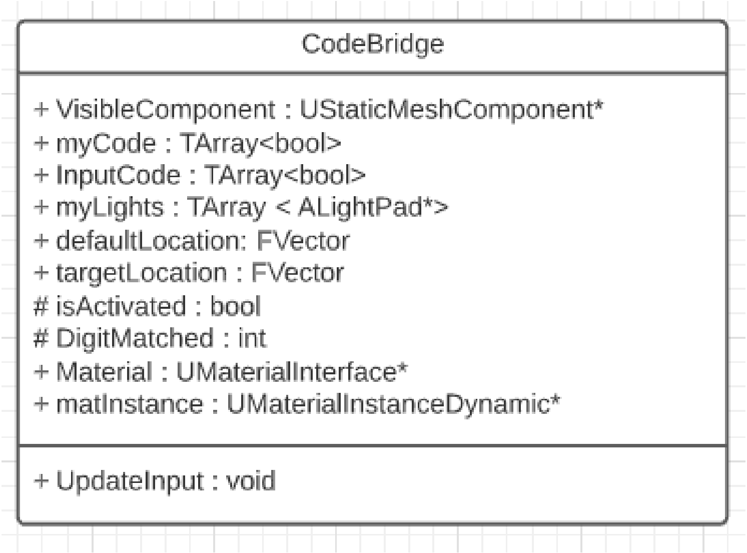
Bridge

+ VisibleComponent : UStaticMeshComponent*
+ mySwitch : AActor*
+ defaultPosition : FVector
+ targetPosition : FVector
# isActivated : bool
+ Material : UMaterialInterface*
+ matInstance : UMaterialInstanceDynamic*

+ ActivateBridge : void

Just a simple actor class with a custom variable mySwitch (AActor*) and during tick function, it checks and cast my switch to the type available and moves from its defaultPosition (FVector) to targetPosition(FVector) when switch is on.

CODEBRIDGE (UE4 ACTOR CLASS)



CodeBridge

+ VisibleComponent : UStaticMeshComponent*
+ myCode : TArray<bool>
+ InputCode : TArray<bool>
+ myLights : TArray < ALightPad*>
+ defaultLocation: FVector
+ targetLocation : FVector
# isActivated : bool
DigitMatched : int
+ Material : UMaterialInterface*
+ matInstance : UMaterialInstanceDynamic*

+ UpdateInput : void

Activated when DigitMatched equals length of its binary code myCode.

UpdateInput is called by member in myLights to update the InputCode for checking.

BLUEPRINT CLASSES

BP_MyGamemode : mainly used for creating blueprint widgets and adding it to screen

HUDs :

- start screen
- pause screen
- end screen
- inventory screen

PET CLASS (UE4 ACTOR CLASS)

- points to owner when first raycasted by player.

- Has states, which can be changed when player releases skills
    - Follow state :
        - Following a distance (followDist) behind player
    - Attract state:
        - Player releases charm skill , will be some distance further from player (charmDist)
        - If player already found by enemy, enemy comes to itself instead of player
    - Heal state:
        - Keep healing player for a constant rate (healRate)
    - Shield state:
        - Ensures player doesn’t receive damage during skill time

- All skills has same cooldown time (skillCD) and is effective for given amount of time (skillTime)
- Boolean skillReady is used by HUD to check and update player when skill is available

- Three colors are available for all three phases of pet chosen by player.
- The three phases are
    - Hungry:
        - When first pick up by player and not fed yet
        - No skills available
    - Healthy:
        - Able to release charm skill, changes to first color
    - Happy:
        - Able to release heal skill, changes to second color
    - Energetic:
        - All three skills unlocked, changes to final color

COMMUNICATION BETWEEN CLASSES

EVENT DISPATCHERS (DELEGATES)

PLAYER CHARACTER

- FMyEventDispatcher (1 parameter, isWinning(bool))
  - Event : GameOverEvent
  - Bound to : EndGameEvent in BP_MYGAMEMODE class which calls out the end screen
  - Broadcasted when : player character health reaches zero and below, player overlap with pittrap and end platform

- FInfoEventDispatcher (3 parameters, name,desc,howTo (FString))
  - Event : ShowInfoEvent
  - Bound to : OnItemInteracted event in BP_MYHUD class which sets the visibility of item information text and content of it based on the inputs when broadcasted.
  - Broadcasted when : Trace function returns item class actor hit

- FMessageDispatcher (1 parameter, myMessage(FString))
  - Event : DisplayMessageEvent
  - Bound to : ShowMessage event in BP_MYHUD class which sets visibility of player message box and set the content to myMessage.
  - Broadcasted when : Trace function hits switches or item is used by player.

- FFlipEventDispatcher (no parameter)
  - Event : FlipSwitchEvent
  - Bound to : ToggleSwitch function in AFLIPSWITCH class, ButtonPressed function in ABUTTONSWITCH class
  - Broadcasted when : player press F when near to FlipSwitch instance or ButtonSwitch instance

- FSkillEventDispatcher(1 paramater,skillNum(int)
  - Event : PetSkillEvent
  - Bound to : useSkill function in APET class
  - Broadcasted when : player press either 1,2,3 and send relative input to pet to release corresponding skill

DOOR KEY (ITEM CLASS)

- FKeyEventDispatcher(no parameters)
  - Event : UnlockEvent
  - Bound to : OnSwitch function in ALOCKEDSWITCH class
  - Broadcasted when : player uses the door key (UseItem function) which unlock the locked switch

MATCH (ITEM CLASS)

- FMatchEventDispatcher(no parameters)
  - Event: LightUpEvent
  - Bound to : ToggleLight function in ATORCHLIGHT class
  - Broadcasted when : player uses the match (UseItem function) near enough to the torchlight

BUTTONSWITCH

- FButtonEventDispatcher (no params)
  - Event: ToggleLightEvent
  - Bound to : ToggleLight function in ALIGHTPAD class
  - Broadcasted when : player pushes button [F] when nearby

FLIPSWITCH

- FFlipSwitchEventDispatcher (no params)
  - Event: ActivateEvent
  - Bound to : ActivateBridge function in ABRIDGE class
  - Broadcasted when : player flips switch [F] when nearby

LOCKEDSWITCH

- FLockSwitchEventDispatcher (no params)
  - Event: ActivateEvent
  - Bound to : ActivateBridge function in ABRIDGE class
  - Broadcasted when : player uses key on it

FUSE

- FFuseEventDispatcher (no params)
  - Event: PowerEvent
  - Bound to : OnPower function in APOWERBOX class
  - Broadcasted when : player uses it when nearby a powerbox

POWERBOX

- FPowerEventDispatcher (no params)
  - Event: PowerSwitchEvent
  - Bound to : unlockSwitch function in AFLIPSWITCH class
  - Broadcasted when : fuse is connected to powerbox

LIGHTPAD

- FCodeEventDispatcher (two params, digit(bool) , index(int))
  - Event: SendCodeEvent
  - Bound to : UpdateInput function in ACODEBRIDGE class which checks input with the actual code
  - Broadcasted when : lightpad is toggled

STUNGRENADE

- FStunEventDispatcher(FVector,grenadeLoc,float,stunTime,float,stunRadius)
  - Event: StunEvent
  - Bound to : updateStun function in ENEMYCHARACTER class which updates Boolean isStun value then used by controller and behavior tree
  - Broadcasted when : stun grenade is placed

PETFOOD

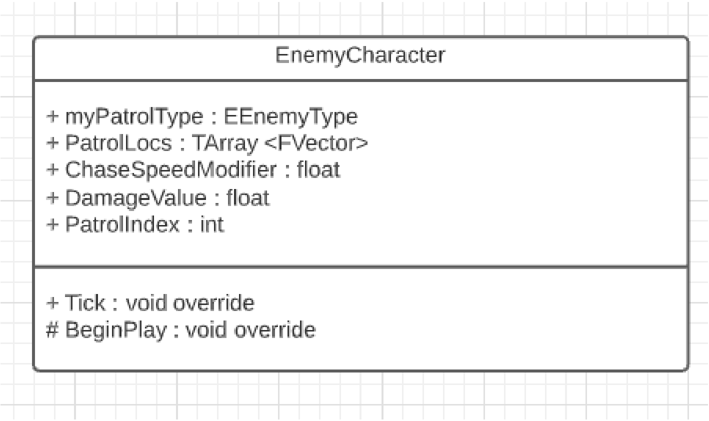
- FFeedEventDispatcher(no param)
  - Event: FeedEvent
  - Bound to : getFood function in PET class which updates myFood in pet to upgrade pet phase
  - Broadcasted when : petfood used by player when pet is owned

PET

- FCharmEventDispatcher (bool,isCharm,FVector,myLoc)
  - Event: CharmEvent
  - Bound to : updateCharmStat function in ENEMYCHARACTER class which updates isCharmed Boolean to be used by behavior tree
  - Broadcasted when : charm skill released by player

AI CLASSES

ENEMY CHARACTER (CHILD OF CHARACTER CLASS)



```
EnemyCharacter

+ myPatrolType : EEnemyType
+ PatrolLocs : TArray <FVector>
+ ChaseSpeedModifier : float
+ DamageValue : float
+ PatrolIndex : int

+ Tick : void override
# BeginPlay : void override
```

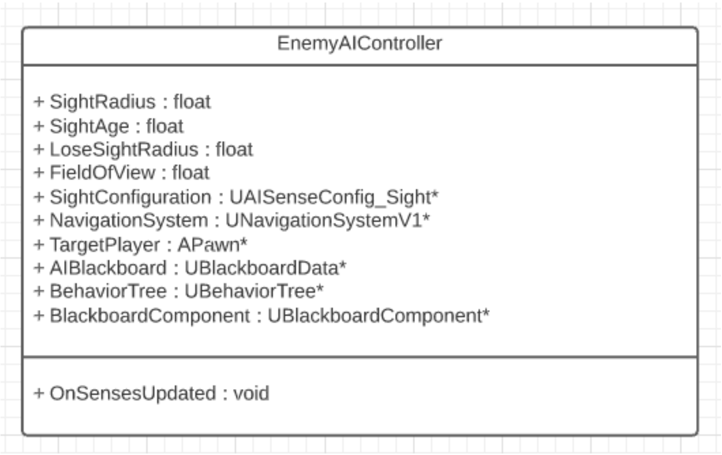
myPatrolType is an enum to decide whether the AI acts as seeker that wanders around or guardian that follows a path around the item it guards.

PatrolLocs include points of the patrol path of guardian while PatrolIndex is used to check which point is it currently in.

ChaseSpeedModifier is the percentage slower than player maximum walk speed.

DamageValue is the amount of damage per tick cause when touching the player.

ENEMY AI CONTROLLER CLASS (CHILD OF AI CONTROLLER CLASS)



```
EnemyAIController

+ SightRadius : float
+ SightAge : float
+ LoseSightRadius : float
+ FieldOfView : float
+ SightConfiguration : UAISenseConfig_Sight*
+ NavigationSystem : UNavigationSystemV1*
+ TargetPlayer : APawn*
+ AIBlackboard : UBlackboardData*
+ BehaviorTree : UBehaviorTree*
+ BlackboardComponent : UBlackboardComponent*

+ OnSensesUpdated : void
```

Sight is used as main perception, all sight related variables are used to set up.

TargetPlayer is used as reference to player-controlled pawn detected by sight.

OnSensesUpdated is used when something is sensed by the perception and to update values in blackboard.

ENEMY AI ADD-ONS

isStunned variable which is used to determined whether stunned and relates to waiting task node in behavior tree.

IsCharmed variable which is used to determined whether pet release charm skills and if they had found player at the same time they will go for pet instead of player.

New type of enemy ai as bomber is added, when spot player, it stands and sends bomb that will follow player and explode in given time.

THANKS FOR READING 😊