

```
struct Point {
    float x, y;
    Point()
    {
        x = y = 0;
    }
    Point(float a, float b)
    {
        x = a, y = b;
    }
};
```

```
CameraDirector

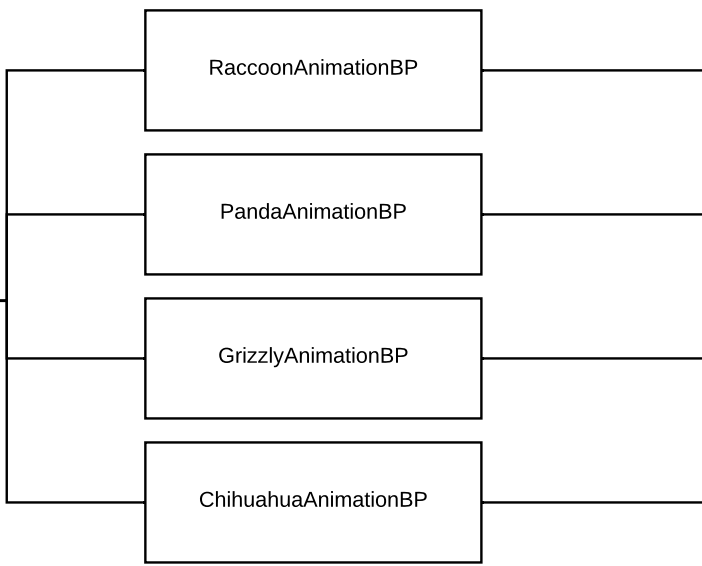
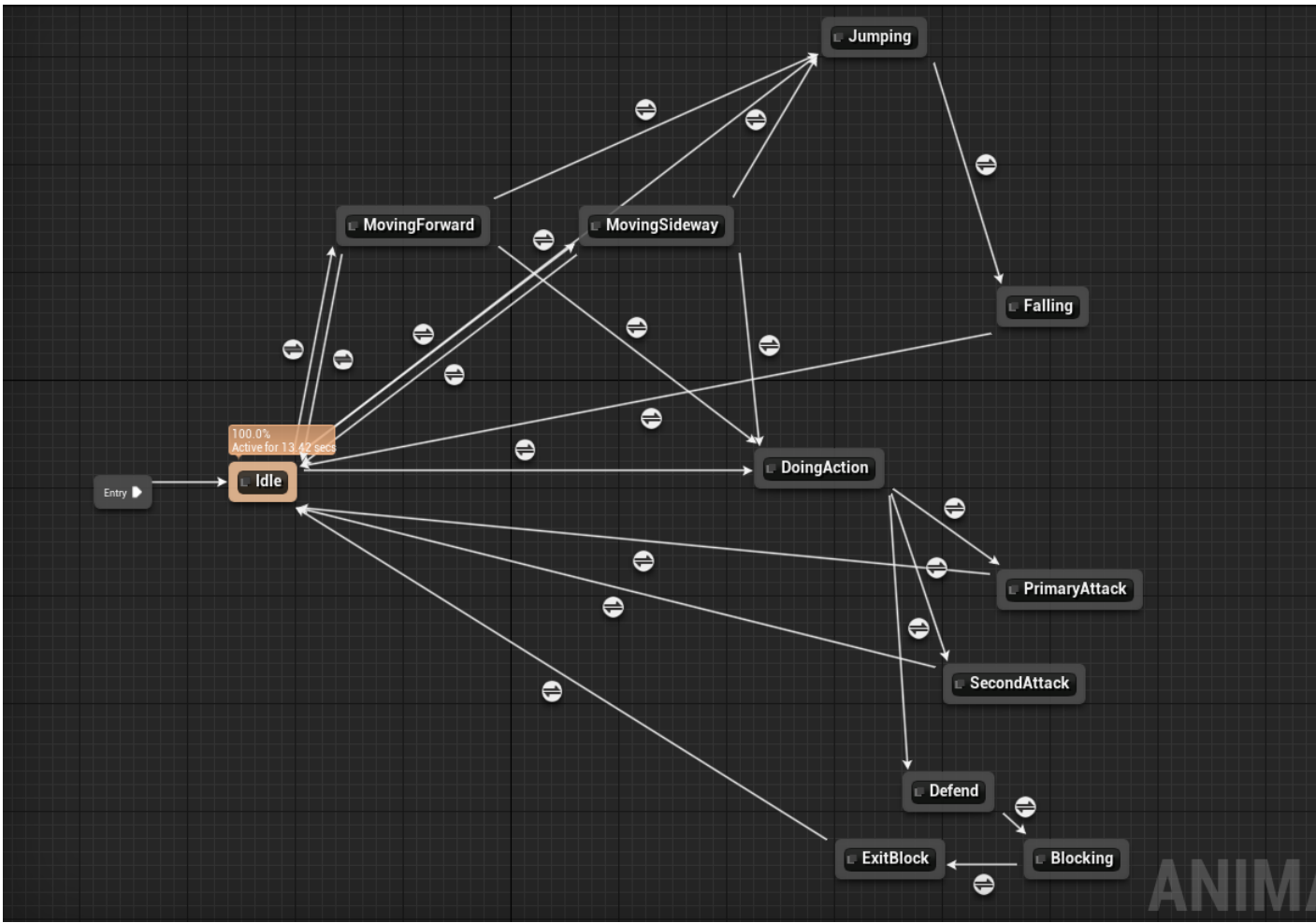
+ CameraOne : AActor*
+ CameraTwo : AActor*
+ NewLocation : FVector
+ NewRotation : FQuat
+ ActorOne : AActor*
+ ActorTwo : AActor*
+ AngleAxis : float
+ Dimensions : FVector
+ AxisVector : FVector
+ Multiplier : float
+ TimeToNextCameraChange : float

- ran : bool = false

+ ACameraDirector()
+ virtual Tick(float DeltaTime) : void <<override>>
+ virtual BeginPlay() : void <<override>>

- slope(float P1, float P2, float Q1, float Q2) : float
- printPoints(Point source, float l, int m) : void
- invertSlope(float slope) : float
```

```
UENUM()
enum class EFighters : uint8
{
    Raccoon UMETA(DisplayName = "Raccoon"),
    Panda UMETA(DisplayName = "Panda"),
    Grizzly UMETA(DisplayName = "Grizzly"),
    Chihuahua UMETA(DisplayName = "Chihuahua")
};
```



```
FighterCharacter

// Animation State Variables
+ primaryAttackStarted : bool
+ secondAttackStarted : bool
+ jumpStarted : bool
+ defendStarted : bool
+ isWalkingForward : bool
+ isWalkingSideway : bool

+ isDoingAction : bool

// Current Fighter State Variable
+ CurrentFighter : EFighters

// Pointer To The Other Fighter For Easier Access
+ FighterOne : AFighterCharacter*
+ FighterTwo : AFighterCharacter*

// References To Skeleton Meshes and Animations Found By The Engine
// Can Be Used For On Stage Fighters
+ UsingEditorToAssign : bool
+ RaccoonSkeletonMesh : class USkeletalMesh*
+ PandaSkeletonMesh : class USkeletalMesh*
+ GrizzlySkeletonMesh : class USkeletalMesh*
+ ChihuahuaSkeletonMesh : class USkeletalMesh*

+ RaccoonAnimationClass : class UClass*
+ PandaAnimationClass : class UClass*
+ GrizzlyAnimationClass : class UClass*
+ ChihuahuaAnimationClass : class UClass*

// AI Related Variables
+ hasBeenDamaged : bool

// TimeHandles (For Animations)
+ PrimaryAttackTimeHandle : FTimerHandle
+ SecondAttackTimeHandle : FTimerHandle
+ MovingForwardTimeHandle : FTimerHandle
+ MovingSidewayTimeHandle : FTimerHandle
+ DoubleJumpTimeHandle : FTimerHandle
+ DefendTimeHandle : FTimerHandle
+ DoingActionTimeHandle : FTimerHandle

// Jump Related
+ DoubleJumpCounter : int
+ JumpHeight : float

// Health System
+ health : float = 100

# virtual BeginPlay() : void <<override>>

// Function For Jump Landing Checking
# virtual Landed(const FHitResult& Hit) : void <<override>>

+ Tick(float DeltaTime) : void <<override>>
+ SetupPlayerInputComponent(class UInputComponent* PlayerInputComponent) : void <<override>>

// Function To Dynamically Set Fighter Type
+ setFighter(EFighters fighter = EFighters::Raccoon) : void

// Damage Helper For Damage Calculation
+ damageHelper(float fullDmg, float blockedDmg) : void

// General Fighter Primary Attack Function
+ virtual stopPrimaryAttack() : void
+ virtual PrimaryAttack() : void

// General Fighter Second Attack Function
+ virtual stopSecondAttack() : void
+ virtual SecondAttack() : void

// General Fighter Moving Forward [X-axis] Function
+ virtual stopMovingForward() : void
+ virtual MovingForward(float Amount) : void

// General Fighter Moving Sideways [Y-axis] Function
+ virtual stopMovingSideway() : void
+ virtual MovingSideway(float Amount) : void

// General Fighter Jump Function
+ virtual DoubleJump() : void

// General Fighter Defend Function
+ virtual stopDefend() : void
+ virtual Defend() : void
+ virtual AIAutoStopDefend() : void

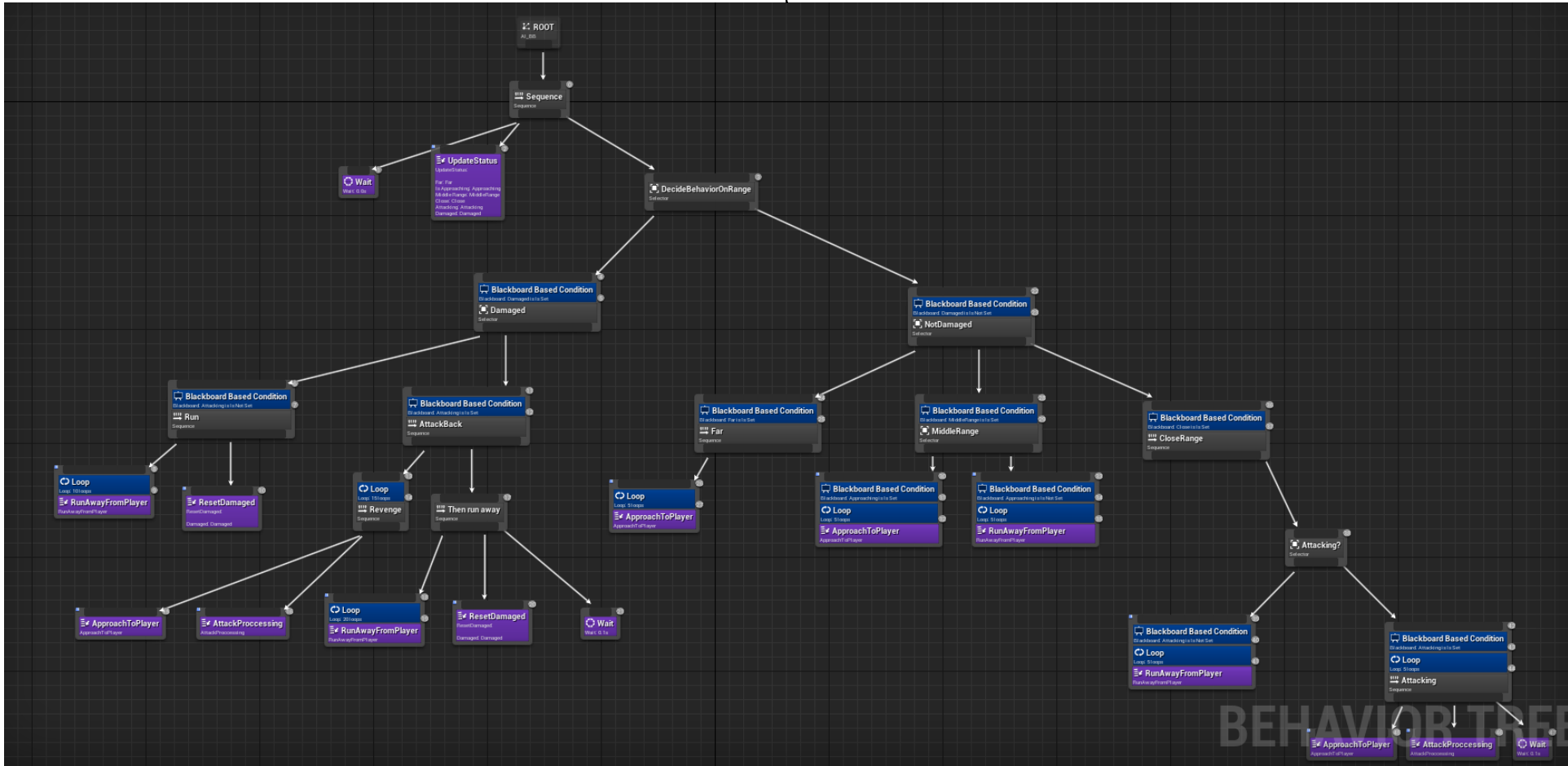
// UI & Public Functions For Health System
+ virtual Damage(float amount) : void
+ virtual MaxHealth() : float
+ virtual getHealth() : float

// Doing Action Handle -> For Cooldowns Among Actions
+ resetDoingAction() : void

// AI Related Function - Determines That Had The Fighter Taken Any Damage Recently
+ hasTakenDamage() : bool
};
```

```
HealthUI

+ updateHealth() : void
```



```
RaccoonCharacter

+ racAtk1AudCue : USoundCue*
+ racAtk1AudComp : UAudioComponent*
+ racAtk2AudCue : USoundCue*
+ racAtk2AudComp : UAudioComponent*
+ racHitAudCue : USoundCue*
+ racHitAudComp : UAudioComponent*
+ racKOAudCue : USoundCue*
+ racKOAudComp : UAudioComponent*
+ doubBellAudCue : USoundCue*
+ doubBellAudComp : UAudioComponent*

+ ARaccoonCharacter(const FObjectInitializer& ObjectInitializer)
+ PostInitializeComponents() <<override>>
+ Tick(float DeltaTime) : void
+ Damage(float amount) : void
+ Defend() : void
```

```
PandaCharacter

+ pandaHitAudCue : USoundCue*
+ pandaHitAudComp : UAudioComponent*
+ pandaJumpAudCue : USoundCue*
+ pandaJumpAudComp : UAudioComponent*
+ pandaKOAudCue : USoundCue*
+ pandaKOAudComp : UAudioComponent*

+ doubBellAudCue : USoundCue*
+ doubBellAudComp : UAudioComponent*

+ APandaCharacter(const FObjectInitializer& ObjectInitializer)
+ PostInitializeComponents() <<override>>
+ Damage(float amount) : void
```