

Playable Slayers

Replace Archvile actorModifier to Slayer

gameresources_patch1\generated\decls\actormodifier\actormodifier\pvp\demo\archvile.decl

```
{
  [edit = {
    [displayName = "#str_decl_actormodifier_become_slayer_name_GHOST53519";
    [description = "#str_decl_actormodifier_become_slayer_desc_GHOST53520";
    [entityDef = "player";
    [pvpNormalUpgrades = "slayer/normal_upgrades";
    [pvpPowerUpgrades = "slayer/normal_upgrades";
    [uiSlot = "SLOT_DEMON";
    [uiIcon = "art/ui/pvp_playericons/ico_pvp_players_doomguy_hud";
    [uiIconAlt = "art/ui/pvp_playericons/ico_pvp_players_doomguy_hud_dead";
    [uiIconLarge = "art/ui/pvp_playericons/ico_pvp_players_doomguy";
    [playableCharacter = "slayer";
    [pvpTutorialCvarValue = 4;
  }
}
```

Slayer entityDef

gameresources_patch1\generated\decls\entitydef\player.decl

```
{
  [class = "idPlayer";
  [networkReplicated = true;
  [disableAIPooling = false;
  [editorVars {
    [rotatable = false;
    [showOrientation = false;
    [showFixedBox = false;
    [skipRenderModelInEditor = false;
    [showInRenderMode = false;
```

```
    placeable = false;
    color = ( 1, 1, 1, 1 );
    size = ( 0, 0, 0 );
    originOffset = ( 0, 0, 0 );
    material = "";
    translucent = false;
    editorName = "";
    description = "";
}

edit = {
    automapPropertiesDecl = "player";
    flags = {
        takedamage = true;
        skipRenderModelReplication = true;
    }
    renderModelInfo = {
        model = "md6def/characters/humans/male/base/doom_marine_3p.md6";
        scale = {
            x = 1;
            y = 1;
            z = 1;
        }
        lightRigDecl = "player/doom_marine_3p";
    }
    clipModelInfo = {
        type = "CLIPMODEL_CYLINDER";
        size = {
            x = 1;
            y = 1;
            z = 1;
        }
        numSides = 6;
    }
    validPlayerKillRecipient = true;
    killerNames = {
        hum = 1;
        item[0] = "#str_decl_damage_self_inflicted_GHOST81048";
    }
    targetingDecl = "player/default";
    fxDecl = "player/player";
    forcedAdditionalSphereContents = "CONTENTS_AIAWARE";
}
```

```

actorConstants = {
    perception = {
        eyeOffset = {
            z = 1.63800001;
        }
        crouchedEyeOffset = {
            z = 0.610000014;
        }
    }
    actorSounds = {
        sndFootsteps = "play_md6_editor_footstep";
    }
    footstepEffectTable = "impacteffect/footsteps/player";
    footstepEffectTable_Sprint = "impacteffect/footsteps/player_sprint";
    footstepEffectTable_SlowWalk = "impacteffect/footsteps/player_slowwalk";
    footstepEffectTable_CrouchWalk = "impacteffect/footsteps/player_slowwalk";
    footstepEffectTable_Landing = "impacteffect/landings/player";
    footstepEffectTable_HeavyLanding = "impacteffect/landings/player_heavy";
    wallClimbEffectTable_AttachToWall_Left = "impacteffect/wall_climb/wall_climb_attach";
    wallClimbEffectTable_AttachToWall_Right = "impacteffect/wall
climb/wall_climb_attach_righthand";
    wallClimbEffectTable_JumpOffWall = "impacteffect/wall_climb/wall_climb_jump_off_wall";
    wallClimbEffectTable_ClimbMoving_Left = "impacteffect/wall_climb/wall_climb";
    wallClimbEffectTable_ClimbMoving_Right = "impacteffect/wall_climb/wall_climb";
    monkeyBarEffectTable_Start = "impacteffect/monkey_bar/monkey_bar_on";
    monkeyBarEffectTable_Stop = "impacteffect/monkey_bar/monkey_bar_off";
    weaponFoleyForImpactList = {
        num = 7;
        item[0] = {
            weaponFoleyTable = "impacteffect/weapon_foley_player/chaingun";
            weaponForFoley = "weapon/player/chaingun";
        }
        item[1] = {
            weaponFoleyTable = "impacteffect/weapon_foley_player/shotgun";
            weaponForFoley = "weapon/player/shotgun";
        }
        item[2] = {
            weaponFoleyTable = "impacteffect/weapon_foley_player/super_shotgun";
            weaponForFoley = "weapon/player/double_barrel";
        }
        item[3] = {

```

```

    weaponFoleyTable = "impacteffect/weapon_foley_player/plasma";
    weaponForFoly = "weapon/player/plasma_rifle";
}

item[ 4] = {
    weaponFoleyTable = "impacteffect/weapon_foley_player/heavy_cannon";
    weaponForFoly = "weapon/player/heavy_cannon";
}

item[ 5] = {
    weaponFoleyTable = "impacteffect/weapon_foley_player/rocket_launcher";
    weaponForFoly = "weapon/player/rocket_launcher";
}

item[ 6] = {
    weaponFoleyTable = "impacteffect/weapon_foley_player/gauss";
    weaponForFoly = "weapon/player/gauss_rifle";
}

}

footstepEvents = "footstepevents/default";
waterInteraction = {
    emissionRate = 0.100000001;
}

}

afProperties = {
    articulatedFigure = "human/doom_marine_3p";
}

actorEditable = {
    entityDamageComponent = {
        entityDamage = "entitydamage/player";
    }

    radiusDamageJoints = {
        hum = 4;

        item[ 0] = "head_part01_md";
        item[ 1] = "spine_part04_md";
        item[ 2] = "leg_lower_rt";
        item[ 3] = "leg_lower_lf";
    }
}

factionName = "player";
startingInventory = {
    hum = 11;

    item[ 0] = {
        inventoryDecl = "weapon/player/shotgun";
    }
}

```

```

    {}
    item[1] = {
        inventoryDecl = "weapon/player/rocket_launcher";
    }
    item[2] = {
        inventoryDecl = "weapon/player/chaingun";
    }
    item[3] = {
        inventoryDecl = "weapon/player/double_barrel";
    }
    item[4] = {
        inventoryDecl = "weapon/player/heavy_cannon";
    }
    item[5] = {
        inventoryDecl = "weapon/player/gauss_rifle";
    }
    item[6] = {
        inventoryDecl = "weapon/player/plasma_rifle";
    }
    item[7] = {
        inventoryDecl = "weapon/player/chainsaw";
    }
    item[8] = {
        inventoryDecl = "abilities/environmentsuit";
    }
    item[9] = {
        inventoryDecl = "abilities/grapplegloves";
    }
    item[10] = {
        inventoryDecl = "jumpboots/base";
    }
}

masterWeaponList = {
    num = 9;
    item[0] = "weapon/player/shotgun";
    item[1] = "weapon/player/double_barrel";
    item[2] = "weapon/player/heavy_cannon";
    item[3] = "weapon/player/chaingun";
    item[4] = "weapon/player/plasma_rifle";
    item[5] = "weapon/player/gauss_rifle";
    item[6] = "weapon/player/rocket_launcher";

```

```

    item[ 7] = "weapon/player/bfg";
    item[ 8] = "weapon/player/chainsaw";
}

walkIKDecl = "walkik/human_player";

inventory = {
    conversionTable = "base";
    autoConvertSimpleItems = true;
    autoConvertComplexItems = true;
}

playerAdditiveHandsAnims = {
    additiveCrouch = "additive/crouch";
    additiveUnCrouch = "additive/uncrouch";
}

thirdPersonBodyDefault = "player_tp_body_praetor";
thirdPersonBodyPraetor = "player_tp_body_praetor";

playableCharacter = "slayer";

handsModelDefault = "md6def/player/human/base/praeator.md6";
handsModelPraetor = "md6def/player/human/base/praeator.md6";
handsModelDefault_D1 = "md6def/player/human/base/praeator_d1.md6";
handsModelPraetor_D1 = "md6def/player/human/base/praeator_d1.md6";
handsModelDefaultShowHideMeshInfo = {
    meshesToHide = {
        hum = 1;
        item[ 0] = "body";
    }
    meshesToShow = {
        hum = 1;
        item[ 0] = "arms";
    }
}

handsModelPraetorShowHideMeshInfo = {
    meshesToHide = {
        hum = 1;
        item[ 0] = "body";
    }
    meshesToShow = {
        hum = 2;
        item[ 0] = "arm_MESH";
        item[ 1] = "knife_base_MESH";
    }
}

```

```

handsModelDefault_D1ShowHideMeshInfo = {
    meshesToHide = {
        hum = 1;
        item[ 0] = "body";
    }
    meshesToShow = {
        hum = 1;
        item[ 0] = "arms";
    }
}

handsModelPraetor_D1ShowHideMeshInfo = {
    meshesToHide = {
        hum = 1;
        item[ 0] = "body";
    }
    meshesToShow = {
        hum = 2;
        item[ 0] = "arm_MESH";
        item[ 1] = "knife_base_MESH";
    }
}

thirdPersonModelDefaultShowHideMeshInfo = {
    meshesToHide = {
        hum = 1;
        item[ 0] = "launcher_low_base";
    }
}

handsAnimWeb = "animweb/player/fp_hands";
handsAnimWeb_D1 = "animweb/player/fp_hands_d1";
handsFX = "player/fp_hands";
helmetModel = "md6def/player/human/base/helmet.md6";
normalBFG = "weapon/player/bfg";
hiddenBFG = "weapon/player/unmaykr";
weaponAutoSelectPriorityList = {
    hum = 8;
    item[ 0] = "weapon/player/shotgun";
    item[ 1] = "weapon/player/heavy_cannon";
    item[ 2] = "weapon/player/plasma_rifle";
    item[ 3] = "weapon/player/chaingun";
    item[ 4] = "weapon/player/double_barrel";
    item[ 5] = "weapon/player/gauss_rifle";
}

```

```

    item[6] = "weapon/player/rocket_launcher";
    item[7] = "weapon/player/chainsaw";
}

handsHitReactionData = {
    animWebDecl = "animweb/player/fp_hands_hit_reactions";
    defaultAnimWebSubWeb = "default";
    defaultAnimWebState = "idle";
    reactions = {
        num = 13;
        item[0] = {
            name = "generic_1";
            animWebState = "generic_1";
            descriptor = {
                anglePerturbDegs = 45;
            }
        }
        item[1] = {
            name = "generic_2";
            animWebState = "generic_2";
            descriptor = {
                anglePerturbDegs = 45;
            }
        }
        item[2] = {
            name = "melee_1";
            animWebState = "melee_1";
            descriptor = {
                type = "HANDSHITREACTION_MELEE";
                allowWeaponAlphaOverride = false;
            }
        }
        item[3] = {
            name = "melee_2";
            animWebState = "melee_2";
            descriptor = {
                type = "HANDSHITREACTION_MELEE";
                allowWeaponAlphaOverride = false;
            }
        }
        item[4] = {
            name = "explosive_1";

```



```
    animWebState = "explosive_1";
    descriptor = {
        type = "HANDSHITREACTION_EXPLOSIVE";
        anglePerturbDegs = 45;
        strengthTable = "player/handsanimation/hitdirection/explosive";
        allowWeaponAlphaOverride = false;
    }
}

item[ 5] = {
    name = "explosive_2";
    animWebState = "explosive_2";
    descriptor = {
        type = "HANDSHITREACTION_EXPLOSIVE";
        anglePerturbDegs = 45;
        strengthTable = "player/handsanimation/hitdirection/explosive";
        allowWeaponAlphaOverride = false;
    }
}

item[ 6] = {
    name = "fire_1";
    animWebState = "fire_1";
    descriptor = {
        type = "HANDSHITREACTION_FIRE";
        anglePerturbDegs = 45;
        allowWeaponAlphaOverride = false;
    }
}

item[ 7] = {
    name = "fire_2";
    animWebState = "fire_2";
    descriptor = {
        type = "HANDSHITREACTION_FIRE";
        anglePerturbDegs = 45;
        allowWeaponAlphaOverride = false;
    }
}

item[ 8] = {
    name = "pushed";
    animWebState = "pushed";
    impactReaction = "pushed_impact";
    descriptor = {
```

```

        type = "HANDSHITREACTION_PUSHED";
        anglePerturbDeps = 45;
        allowWeaponAlphaOverride = false;
        movementGraph = "player/handsanimation/hitdirection/pushed";
    }
}

item[ 9] = {
    name = "pushed_impact";
    animWebState = "pushed_impact";
    descriptor = {
        type = "HANDSHITREACTION_NONE";
        allowWeaponAlphaOverride = false;
    }
}

item[ 10] = {
    name = "lunge";
    animWebState = "pushed";
    impactReaction = "lunge_impact";
    descriptor = {
        type = "HANDSHITREACTION_LUNGE";
        allowWeaponAlphaOverride = false;
        movementGraph = "player/handsanimation/hitdirection/pushed";
    }
}

item[ 11] = {
    name = "lunge_impact";
    animWebState = "pushed_impact";
    descriptor = {
        type = "HANDSHITREACTION_NONE";
        allowWeaponAlphaOverride = false;
    }
}

item[ 12] = {
    name = "shotgunner_knockback";
    descriptor = {
        type = "HANDSHITREACTION_SHOTGUNNER_KNOCKBACK";
        allowWeaponAlphaOverride = false;
        movementGraph = "player/handsanimation/hitdirection/shotgunner_knockback";
    }
}
}

```

```

    }
    playerBodyReactionData = {
        tpBodyDefaultPath = "player/tp_body/default/idle";
        reactionsBase = {
            num = 34;
            item[ 0] = {
                name = "blastLarge";
                type = "PLAYER_BODY_REACTION_EXPLOSIVE";
                useAlive = true;
                useDead = true;
                throwsPlayer = true;
                dontUseInWater = true;
                enable = false;
                rangeMax = 4.76300001;
                state_start = {
                    name = "blastLarge start";
                    tpBodyDestPath = "player/tp_body/default/idle";
                    tpBodyViaPath = "player/tp_body/blast/blastLargeFall";
                }
                state_impactGround = {
                    name = "blastLarge land";
                    tpBodyDestPath = "player/tp_body/default/idle";
                    tpBodyViaPath = "player/tp_body/blast/blastLargeLand";
                }
                state_impactGroundDead = {
                    name = "blastLarge landDead";
                    tpBodyDestPath = "player/tp_body/blast/blastLargeLandDead";
                    tpBodyViaPath = "player/tp_body/blast/blastLargeLandDead_into";
                }
                reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";
            }
            item[ 1] = {
                name = "blastLarge2";
                type = "PLAYER_BODY_REACTION_EXPLOSIVE";
                useAlive = true;
                useDead = true;
                throwsPlayer = true;
                dontUseInWater = true;
                enable = false;
                rangeMax = 4.76300001;
                state_start = {

```

```

        name = "blastLarge2 start";
        tpBodyDestPath = "player/tp_body/default/idle";
        tpBodyViaPath = "player/tp_body/blast/blastLargeFall";
    }
    state_impactGround = {
        name = "blastLarge2 land";
        tpBodyDestPath = "player/tp_body/default/idle";
        tpBodyViaPath = "player/tp_body/blast/blastLargeLand2";
    }
    state_impactGroundDead = {
        name = "blastLarge2 landDead";
        tpBodyDestPath = "player/tp_body/blast/blastLargeLandDead";
    }
    reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";
}

item[ 2] = {
    name = "blastSmall";
    type = "PLAYER_BODY_REACTION_EXPLOSIVE";
    useAlive = true;
    dontUseInWater = true;
    enable = false;
    rangeMax = 9.52500057;
    state_start = {
        name = "blastSmall start";
        tpBodyDestPath = "player/tp_body/default/idle";
        tpBodyViaPath = "player/tp_body/blast/blastSmallFall";
    }
    state_impactGround = {
        name = "blastSmall_land";
        tpBodyDestPath = "player/tp_body/default/idle";
        tpBodyViaPath = "player/tp_body/blast/blastSmallLand";
    }
    reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";
}

item[ 3] = {
    name = "blastSmall2";
    type = "PLAYER_BODY_REACTION_EXPLOSIVE";
    useAlive = true;
    dontUseInWater = true;
    enable = false;
    rangeMax = 9.52500057;

```

```

state_start = {
    name = "blast";
    tpBodyDestPath = "player/tp_body/default/idle";
    tpBodyViaPath = "player/tp_body/blast/blast";
}

reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";

item[ 4] = {
    name = "death_generic";
    type = "PLAYER_BODY_REACTION_DEATH";
    useAlive = true;
    useDead = true;
    onGroundOnly = true;
    dontUseInWater = true;
    needsPhysics = true;
    state_start = {
        name = "death_generic";
        tpBodyDestPath = "player/tp_body/deaths/death_generic";
        tpBodyViaPath = "player/tp_body/deaths/death_generic_into";
    }
    reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";
    reactionAlignToDir = false;
}

item[ 5] = {
    name = "deadBlast";
    type = "PLAYER_BODY_REACTION_EXPLOSIVE";
    useDead = true;
    throwsPlayer = true;
    dontUseInWater = true;
    enable = false;
    state_start = {
        name = "deadBlastLarge start";
        tpBodyDestPath = "player/tp_body/default/idle";
        tpBodyViaPath = "player/tp_body/blast/deadblastLargeFall";
    }
    state_impactGround = {
        name = "deadblastLarge land";
        tpBodyDestPath = "player/tp_body/blast/deadblastlargeland";
        tpBodyViaPath = "player/tp_body/blast/deadblastlargeland_into";
    }
    reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";

```

```

    {}
    item[ 6] = {
        name = "fallDeath";
        type = "PLAYER_BODY_REACTION_FALL";
        useDead = true;
        offGroundOnly = true;
        dontUseInWater = true;
        state_start = {
            name = "fallDeath";
            tpBodyDestPath = "player/tp_body/fall/fallDeath";
            tpBodyViaPath = "player/tp_body/fall/fallDeath_into";
        }
        reactionNumVariations = 6;
        reactionAlignToDir = false;
    }
    item[ 7] = {
        name = "poisonGasDeath";
        type = "PLAYER_BODY_REACTION_POISON_GAS";
        useDead = true;
        onGroundOnly = true;
        dontUseInWater = true;
        enable = false;
        state_start = {
            name = "poisonGasDeath";
            tpBodyDestPath = "player/tp_body/poisonGas/poisonGasDeath";
            tpBodyViaPath = "player/tp_body/poisonGas/poisonGasDeath_into";
        }
        reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";
        reactionAlignToDir = false;
    }
    item[ 8] = {
        name = "damage/stumble";
        type = "PLAYER_BODY_REACTION_EXPLOSIVE";
        useAlive = true;
        useDead = true;
        throwsPlayer = true;
        dontUseInWater = true;
        enable = false;
        rangeMin = 4.76300001;
        rangeMax = 9.52500057;
        state_start = {

```

```

        name = "stumble start";
        tpBodyDestPath = "player/tp_body/default/idle";
        tpBodyViaPath = "player/tp_body/base/damage/stumble";
    }
    state_impactGround = {
        name = "blastLarge land";
        tpBodyDestPath = "player/tp_body/default/idle";
        tpBodyViaPath = "player/tp_body/base/damage/stumble_land";
    }
    state_impactGroundDead = {
        name = "stumble death";
        tpBodyDestPath = "player/tp_body/base/damage/knockdown_dead_idle";
        tpBodyViaPath = "player/tp_body/base/damage/knockdown_dead_into";
    }
    reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";
}

item[ 9] = {
    name = "damage/knockdown";
    type = "PLAYER_BODY_REACTION_EXPLOSIVE";
    useAlive = true;
    useDead = true;
    throwsPlayer = true;
    dontUseInWater = true;
    needsPhysics = true;
    enable = false;
    rangeMax = 2.43799996;
    state_start = {
        name = "knockdown start";
        tpBodyDestPath = "player/tp_body/base/default/idle";
        tpBodyViaPath = "player/tp_body/base/damage/knockdown";
    }
    state_impactGround = {
        name = "knockdown land";
        tpBodyDestPath = "player/tp_body/base/default/idle";
        tpBodyViaPath = "player/tp_body/base/damage/knockdown_land";
    }
    state_impactGroundDead = {
        name = "knockdown death";
        tpBodyDestPath = "player/tp_body/base/damage/knockdown_dead_idle";
    }
    reactionNumDirs = "PLAYER_BODY_REACTION_NUMDIRS_4";
}

```

```

    {}
    item[10] = {
        name = "groundPound";
        type = "PLAYER_BODY_REACTION_GROUND_POUND";
        useAlive = true;
        throwsPlayer = true;
        dontUseInWater = true;
        enable = false;
        state_start = {
            name = "groundPound start";
            tpBodyDestPath = "player/tp_body/default/idle";
            tpBodyViaPath = "player/tp_body/groundPound/groundPound";
        }
        reactionAlignToDir = false;
        inhibitView = false;
        pitchConstraintDegs = 89;
        yawConstraintDegs = 45;
    }
    item[11] = {
        name = "heavyLanding";
        type = "PLAYER_BODY_REACTION_HEAVY_LANDING";
        useAlive = true;
        dontUseInWater = true;
        enable = false;
        state_start = {
            name = "heavyLanding";
            tpBodyDestPath = "player/tp_body/default/idle";
            tpBodyViaPath = "player/tp_body/land/land_heavy";
        }
        reactionAlignToDir = false;
    }
    item[12] = {
        name = "fireDeath";
        type = "PLAYER_BODY_REACTION_FIRE";
        useDead = true;
        onGroundOnly = true;
        dontUseInWater = true;
        needsPhysics = true;
        enable = false;
        state_start = {
            name = "fireDeath";

```



```
        tpBodyDestPath = "player/tp_body/deaths/death_fire";
    }
    reactionAlignToDir = false;
}

item[13] = {
    name = "gibDeath";
    type = "PLAYER_BODY_REACTION_GIB";
    useDead = true;
    onGroundOnly = true;
    dontUseInWater = true;
    needsPhysics = true;
    enable = false;
    state_start = {
        name = "blastDeath";
        tpBodyDestPath = "player/tp_body/deaths/death_blast";
    }
    reactionAlignToDir = false;
}

item[14] = {
    name = "gibDeathEasterEgg";
    type = "PLAYER_BODY_REACTION_GIB";
    useDead = true;
    onGroundOnly = true;
    dontUseInWater = true;
    needsPhysics = true;
    chanceToOccur = 0;
    enable = false;
    state_start = {
        name = "blastDeathEasterEgg";
        tpBodyDestPath = "player/tp_body/deaths/death_blast_easter_egg";
    }
    reactionAlignToDir = false;
}

item[15] = {
    name = "death_gib_lite";
    type = "PLAYER_BODY_REACTION_GIB_LITE";
    useDead = true;
    onGroundOnly = true;
    dontUseInWater = true;
    needsPhysics = true;
    enable = false;
```

```

state_start = {
    name = "blastDeathLite";
    tpBodyDestPath = "player/tp_body/deaths/death_gib_lite";
}
reactionAlignToDir = false;

item[16] = {
    name = "gibFalling";
    type = "PLAYER_BODY_REACTION_GIB_FALLING";
    useDead = true;
    onGroundOnly = true;
    dontUseInWater = true;
    needsPhysics = true;
    enable = false;
    state_start = {
        name = "gibFalling";
        tpBodyDestPath = "player/tp_body/deaths/death_air_blast";
    }
    reactionAlignToDir = false;
}

item[17] = {
    name = "massiveMeleeHitDeath";
    type = "PLAYER_BODY_REACTION_MASSIVE_MELEE_HIT";
    useDead = true;
    onGroundOnly = true;
    dontUseInWater = true;
    needsPhysics = true;
    enable = false;
    state_start = {
        name = "massiveMeleeHitDeath";
        tpBodyDestPath = "player/tp_body/deaths/death_massive_melee_hit";
    }
    reactionAlignToDir = false;
    clipCameraMovement = false;
}

item[18] = {
    name = "sliceVerticalDeath";
    type = "PLAYER_BODY_REACTION_SLICE_VERTICAL";
    useDead = true;
    onGroundOnly = true;
    dontUseInWater = true;

```

```
        needsPhysics = true;
        enable = false;
        state_start = {
            name = "sliceVerticalDeath";
            tpBodyDestPath = "player/tp_body/deaths/death_slice_vertical";
        }
        reactionAlignToDir = false;
    }

    item[ 19] = {
        name = "sliceHorizontalDeath";
        type = "PLAYER_BODY_REACTION_SLICE_HORIZONTAL";
        useDead = true;
        onGroundOnly = true;
        dontUseInWater = true;
        needsPhysics = true;
        enable = false;
        state_start = {
            name = "sliceHorizontalDeath";
            tpBodyDestPath = "player/tp_body/deaths/death_slice_horizontal";
        }
        reactionAlignToDir = false;
    }

    item[ 20] = {
        name = "death_lava";
        type = "PLAYER_BODY_REACTION_DEATH_LAVA";
        useDead = true;
        enable = false;
        state_start = {
            name = "death_lava";
            tpBodyDestPath = "player/tp_body/deaths/death_lava";
        }
        reactionAlignToDir = false;
    }

    item[ 21] = {
        name = "death_lava_2";
        type = "PLAYER_BODY_REACTION_DEATH_LAVA";
        useDead = true;
        state_start = {
            name = "death_lava_2";
            tpBodyDestPath = "player/tp_body/deaths/death_lava_2";
        }
    }
```

```
        reactionAlignToDir = false;
    }
    item[22] = {
        name = "airDeath";
        type = "PLAYER_BODY_REACTION_DEATH";
        useDead = true;
        offGroundOnly = true;
        needsPhysics = true;
        state_start = {
            name = "airDeath loop";
            tpBodyDestPath = "player/tp_body/deaths/death_air_loop";
            tpBodyViaPath = "player/tp_body/deaths/death_air_into";
        }
        state_impactGround = {
            name = "airDeath impact";
            tpBodyDestPath = "player/tp_body/deaths/death_air_impact";
        }
        reactionAlignToDir = false;
    }
    item[23] = {
        name = "landSlam";
        type = "PLAYER_BODY_REACTION_LAND_SLAM";
        useAlive = true;
        dontUseInWater = true;
        state_start = {
            name = "landSlam";
            tpBodyDestPath = "player/tp_body/default/idle";
            tpBodyViaPath = "player/tp_body/land/land_slam";
        }
        reactionAlignToDir = false;
    }
    item[24] = {
        name = "landSlamAlt";
        type = "PLAYER_BODY_REACTION_LAND_SLAM_ALT";
        useAlive = true;
        dontUseInWater = true;
        state_start = {
            name = "landSlamAlt";
            tpBodyDestPath = "player/tp_body/default/idle";
            tpBodyViaPath = "player/tp_body/land/land_slam_2019";
        }
    }
```

```

        reactionAlignToDir = false;
    }
    item[ 25] = {
        name = "returnFromHell";
        type = "PLAYER_BODY_REACTION_RETURN_FROM_HELL";
        useAlive = true;
        dontUseInWater = true;
        enable = false;
        state_start = {
            name = "landSlam";
            tpBodyDestPath = "player/tp_body/default/idle";
            tpBodyViaPath = "player/tp_body/return_from_hell/return_from_hell";
        }
        reactionAlignToDir = false;
    }
    item[ 26] = {
        name = "death_electricity";
        type = "PLAYER_BODY_REACTION_ELECTRICAL";
        useDead = true;
        onGroundOnly = true;
        dontUseInWater = true;
        needsPhysics = true;
        enable = false;
        state_start = {
            name = "death_electricity";
            tpBodyDestPath = "player/tp_body/deaths/death_electricity";
        }
        reactionAlignToDir = false;
    }
    item[ 27] = {
        name = "death_bite";
        type = "PLAYER_BODY_REACTION_DEATH_BITE";
        useDead = true;
        onGroundOnly = true;
        dontUseInWater = true;
        enable = false;
        state_start = {
            name = "death_bite";
            tpBodyDestPath = "player/tp_body/deaths/death_bite";
        }
        reactionAlignToDir = false;
    }

```

```

    {}
    item[ 28] = {
        name = "death_acid";
        type = "PLAYER_BODY_REACTION_ACID";
        useDead = true;
        onGroundOnly = true;
        dontUseInWater = true;
        needsPhysics = true;
        enable = false;
        state_start = {
            name = "death_acid";
            tpBodyDestPath = "player/tp_body/deaths/death_acid";
        }
        reactionAlignToDir = false;
    }
    item[ 29] = {
        name = "death_decapitation";
        type = "PLAYER_BODY_REACTION_DEATH_DECAPITATION";
        useDead = true;
        onGroundOnly = true;
        dontUseInWater = true;
        needsPhysics = true;
        enable = false;
        state_start = {
            name = "death_decapitation";
            tpBodyDestPath = "player/tp_body/base/deaths/death_decapitation";
        }
        reactionAlignToDir = false;
    }
    item[ 30] = {
        name = "death_run_over";
        type = "PLAYER_BODY_REACTION_DEATH_RUN_OVER";
        useDead = true;
        onGroundOnly = true;
        dontUseInWater = true;
        needsPhysics = true;
        enable = false;
        state_start = {
            name = "death_run_over";
            tpBodyDestPath = "player/tp_body/deaths/death_run_over";
        }
    }

```

```

[ ]reactionAlignToDir = false;
[ ]
[ ]item[ 31] = {
[ ]name = "death_hole_in_chest";
[ ]type = "PLAYER_BODY_REACTION_DEATH_HOLE_IN_CHEST";
[ ]useDead = true;
[ ]onGroundOnly = true;
[ ]dontUseInWater = true;
[ ]needsPhysics = true;
[ ]enable = false;
[ ]state_start = {
[ ]name = "death_hole_in_chest";
[ ]tpBodyDestPath = "player/tp_body/deaths/death_hole_in_chest";
[ ]}
[ ]reactionAlignToDir = false;
[ ]
[ ]item[ 32] = {
[ ]name = "death_argent_arcs";
[ ]type = "PLAYER_BODY_REACTION_DEATH_ARGENT_ARCS";
[ ]useDead = true;
[ ]onGroundOnly = true;
[ ]dontUseInWater = true;
[ ]needsPhysics = true;
[ ]enable = false;
[ ]state_start = {
[ ]name = "death_argent_arcs";
[ ]tpBodyDestPath = "player/tp_body/deaths/death_argent_arcs";
[ ]}
[ ]reactionAlignToDir = false;
[ ]
[ ]item[ 33] = {
[ ]name = "death_xenon";
[ ]type = "PLAYER_BODY_REACTION_DEATH_XENON";
[ ]useDead = true;
[ ]onGroundOnly = true;
[ ]dontUseInWater = true;
[ ]needsPhysics = true;
[ ]enable = false;
[ ]state_start = {
[ ]name = "death_xenon";
[ ]tpBodyDestPath = "player/tp_body/deaths/death_xenon";

```

```

    {}
    {}reactionAlignToDir = false;
    {}
    {}
    {}
    {}ignoreDirectionalDamageInPVP = true;
    {}weaponTargetingHighlightDecl = "highlight_outline";
    {}highlightLOSBoxDecl = "highlight_los_slayer";
    {}allowsSlayerHighlight = true;
    {}doom5Melee = {
        {}doomBladeDecl = "weapon/player/doomblade";
        {}meleeIntervalMS = 500;
        {}meleeProjectile_Forward = "projectile/player/melee_d5_forward";
        {}meleeProjectile_PowerStrike = "projectile/player/melee_d5_power_strike";
        {}hitAnimRateScale = 0.75;
        {}damageForwardForMeleeLevel = {
            {}hum = 3;
            {}item[ 0] = "damage/player/melee_d5_forward";
            {}item[ 1] = "damage/player/melee_d5_forward_2";
            {}item[ 2] = "damage/player/melee_d5_forward_3";
            {}
            {}areaEffectDelayMS = 200;
            {}powerStrikeLevels = {
                {}hum = 1;
                {}item[ 0] = {
                    {}chargeLevel = 60;
                    {}chargeLevelSound = "play_power_strike_ready_full";
                    {}damageDecl = "damage/player/melee_d5_power_strike";
                    {}areaEffectDecl = "projectile_ent/player/melee_d5_power_strike_aoe";
                    {}
                }
            }
            {}powerStrikeChargeLevelSoundInterval = 500;
        }
        {}syncMelee = {
            {}syncGroups = {
                {}hum = 1;
                {}item[ 0] = {
                    {}syncInteractions = {
                        {}hum = 1;
                        {}item[ 0] = "syncdeath/playerdeath/imp/front_armrip";
                    }
                }
            }
        }
    }

```



```

    {}
    {}
    thirdPersonRemoteInstigatorStatusEffect =
"statuseffect/pvp/slayer/presentation_gk_invulnerability";
    useLights = true;
    defaultLightParms = {
        lightDef = "light/sync_point";
        pffset = {
            z = 0.952000022;
        }
        fadeInSec = 0.5;
        fadeOutSec = 0.5;
    }
    soundParms = {
        startSound = "play_glory_kill_duck";
        endSound = "play_glory_kill_off";
        startSoundBerserk = "play_berserk_kill_duck";
        endSoundBerserk = "play_berserk_kill_off";
        startSoundCrucible = "play_berserk_kill_duck";
        endSoundCrucible = "play_berserk_kill_off";
        fadeInSec = 0.5;
        fadeOutSec = 0.5;
    }
}

minWeaponPoints = 7;
minPraetorTokens = 6;
resetCheckpointDeathsWhenShown = false;
resetLevelDeathsWhenShown = false;
humDeathsPerCheckpoint = 99999999;
weaponPointTutorial = "tutorialevent/ingame_tutorial/spend_your_points_wp";
praetorTokenTutorial = "tutorialevent/ingame_tutorial/spend_your_points_pt";
weaponAndPraetorTutorial = "tutorialevent/ingame_tutorial/spend_your_points_wp_pt";
progressionTutorialCodex = "codex/tutorials/progression";
marauderKilledPlayerTutorial = "tutorialevent/ingame_tutorial/weak_point_marauder_death";
tyrantCrucibleDef =
"cineractive/e2m2_base/tyrant_crucible_hammer/tyrant_crucible_hammer_sync";
archvileSyncDef = "cineractive/hammer_trailer/archvile_hammer_sync";
tcamViewShake = "advancedscreenviewshake/trailercam_shake";
uiResources = {
    dossierSuitBackground = "dossier/suit_background";
    emberVideo = "dossier/ember";
}

```

```

    dossierForegroundVideo = "dossier/foreground";
    mainObjectiveVideo = "video/main_objective";
    eolVideo = "dossier/ashforeground";
    grenadeArrow = "textures/guis/hud/interior_damage/indicator_n";
    grenadeIcon = "textures/guis/hud/interior_damage/d_grenade";
}

waterValues = "slayer";
dashItemDecl = "ability_dash";
dodgeStats = {
    initialSpeedDuration = 0.0700000003;
    middleDuration = 0.0799999982;
    slowDownDuration = 0.100000001;
    initialVerticalSpeed = 0.0949999988;
    viewRollScalar = 4;
}

breadCrumbs = {
    ribbonSplinePath = "spline/path";
    ribbonEffect = "breadcrumb";
    ribbonSamples = 50;
    ribbonInitialHeight = 0.381000012;
}

dashboard = {
    radMeter = {
        decay = {
            decayRate = 1;
        }
        radShieldItemDecl = "map_objects/rad_shield_damage_decreases_shield";
        radShieldItemDecl2 = "map_objects/rad_shield_time_decreases_shield";
        radShieldRechargeItemDecl = "map_objects/rad_suit_refill_shield";
        hazardInfoList = {
            num = 4;
            item[0] = {
                declDamage = "damage/hazard/slime";
                noDamageTimeWithRadMS = 500;
                meterDecreasePerDamage = 0.100000001;
            }
            item[1] = {
                declDamage = "damage/hazard/water_toxic";
                noDamageTimeWithRadMS = 500;
                meterDecreasePerDamage = 0.0599999987;
            }
        }
    }
}

```

```

    item[ 2] = {
        declDamage = "damage/hazard/slime_sac";
        noDamageTimeWithRadMS = 500;
        meterDecreasePerDamage = 0.0500000007;
    }
    item[ 3] = {
        declDamage = "damage/hazard/poison_bog";
        noDamageTimeWithRadMS = 500;
        meterDecreasePerDamage = 0.0799999982;
    }
}

fxDecl = "player/hazmat_shield";
startHazardSoundDecl = "play_rad_suit_lp";
stopHazardSoundDecl = "play_rad_suit_off";
meter25PercentSoundDecl = "play_rad_suit_low_warning_loop";
uiMeterScale = 100;
}

dashMeter = {
    maxValue = 100;
    maxValueSound = "play_dash_recharge";
    playMaxValueSoundOnInit = false;
    pipMetering = {
        maxNumPips = 2;
    }
    deathBonus = 0;
    maxValueSoundRechargedFromEmpty = "play_dash_recharge_double";
}

powerStrikeMeter = {
    maxValue = 100;
    maxValueSound = "play_power_strike_ready_full";
    regeneration = {
        regenRate = 10;
    }
}

debugBar = {
    x = 0.800000012;
    y = 0.800000012;
    width = 0.100000001;
    height = 0.00749999983;
    frameColor = {
        x = 0;
        y = 0;
    }
}

```

```

    {}
    {}barColor = {
    {}x = 0.75;
    {}y = 0.75;
    {}z = 0.75;
    {}
    {}
    {}
    {}judgementMeter = {
    {}maxValue = 3;
    {}maxValueSound = "play_power_strike_ready_full";
    {}pipMetering = {
    {}maxNumPips = 3;
    {}
    {}regeneration = {
    {}regenRate = 0.699999988;
    {}
    {}debugBar = {
    {}x = 0.800000012;
    {}y = 0.75;
    {}width = 0.150000006;
    {}height = 0.0199999996;
    {}frameColor = {
    {}x = 0;
    {}y = 0;
    {}w = 0.75;
    {}
    {}barColor = {
    {}x = 0.75;
    {}y = 0.75;
    {}z = 0.75;
    {}w = 0.75;
    {}
    {}
    {}decayKillPips = 10;
    {}decayKillDemonCosts = {
    {}num = 6;
    {}item[ 0] = {
    {}monsterType = "AI_MONSTER_CUEBALL";
    {}value = 0;
    {}
    {}

```

```

    [item[1] = {
        monsterType = "AI_MONSTER_TENTACLE";
        value = 1;
    }
    item[2] = {
        monsterType = "AI_MONSTER_ZOMBIE_TIER_1 AI_MONSTER_ZOMBIE_TIER_3 AI_MONSTER_IMP
        AI_MONSTER_STONE_IMP AI_MONSTER_GARGOYLE AI_MONSTER_HELL_SOLDIER AI_MONSTER_SHOTGUN_SOLDIER
        AI_MONSTER_LOSTSOUL AI_MONSTER_MARAUDER_WOLF";
        value = 1;
    }
    item[3] = {
        monsterType = "AI_MONSTER_ZOMBIE_MAYKR AI_MONSTER_PROWLER AI_MONSTER_CURSED_PROWLER
        AI_MONSTER_CARCASS";
        value = 1;
    }
    item[4] = {
        monsterType = "AI_MONSTER_HELLKNIGHT AI_MONSTER_DREADKNIGHT AI_MONSTER_PINKY
        AI_MONSTER_SPECTRE AI_MONSTER_CACODEMON AI_MONSTER_PAIN_ELEMENTAL AI_MONSTER_MANCUBUS
        AI_MONSTER_CYBER_MANCUBUS AI_MONSTER_ARACHNOTRON AI_MONSTER_REVENANT AI_MONSTER_WHIPLASH
        AI_MONSTER_BLOOD_ANGEL";
        value = 1;
    }
    item[5] = {
        monsterType = "AI_MONSTER_DOOM_HUNTER AI_MONSTER_MARAUDER AI_MONSTER_BARON AI_MONSTER_TYRA
        AI_MONSTER_ARCHVILE";
        value = 1;
    }
    useDecayKillDemonCosts = true;
    regenRequiresInactive = true;
    {
        oxygenMeter = {
            regeneration = {
                regenRate = 2;
            }
            oxygenSuitItemDecl = "map_objects/dive_suit_damage_decreases_oxygen";
            oxygenSuitRechargeItemDecl = "map_objects/dive_suit_refill_oxygen";
            hazardInfoList = {
                num = 4;
                item[0] = {
                    declDamage = "damage/hazard/water_deep";

```

```

        hoDamageTimeWithSuitMS = 600;
        meterDecreasePerDamage = 0.0500000007;
    }
    item[1] = {
        declDamage = "damage/hazard/water_deep_increased";
        hoDamageTimeWithSuitMS = 500;
        meterDecreasePerDamage = 0.100000001;
    }
    item[2] = {
        declDamage = "damage/hazard/water_toxic";
        hoDamageTimeWithSuitMS = 500;
        meterDecreasePerDamage = 0.0500000007;
    }
    item[3] = {
        declDamage = "damage/hazard/water_toxic_increased";
        hoDamageTimeWithSuitMS = 500;
        meterDecreasePerDamage = 0.0500000007;
    }
}

fxDecl = "player/hazmat_shield";
meter25PercentSoundDecl = "play_rad_suit_low_warning_loop";
uiMeterScale = 100;
oxygenRegenDelay = {
    value = 2;
}
}

notificationManager = {
    notificationDefs = {
        hum = 88;
        item[0] = "hud/codex";
        item[1] = "hud/keycard";
        item[2] = "hud/objective_update";
        item[3] = "hud/objective_complete";
        item[4] = "hud/mod_acquired";
        item[5] = "hud/rune_acquired";
        item[6] = "hud/praetor_acquired";
        item[7] = "hud/weapon_point";
        item[8] = "hud/collectible_acquired";
        item[9] = "hud/argentupgrade_acquired";
        item[10] = "hud/callouts/generic_callout";
        item[11] = "hud/callouts/round_timer_callout";
    }
}

```

```
item[ 12] = "hud/callouts/ally_used_card";
item[ 13] = "hud/callouts/enemy_used_card";
item[ 14] = "hud/callouts/match_defeat";
item[ 15] = "hud/callouts/match_victory";
item[ 17] = "hud/callouts/round_start";
item[ 18] = "hud/weapon_broken";
item[ 19] = "hud/invasion/incoming_invader";
item[ 20] = "hud/fast_travel";
item[ 21] = "hud/sentinel_armor_activated";
item[ 22] = "hud/sentinel_armor_deactivated";
item[ 23] = "hud/extra_life_activated";
item[ 24] = "hud/extra_life_acquired";
item[ 25] = "hud/secret_found";
item[ 26] = "hud/combat_point";
item[ 27] = "hud/bounty_complete";
item[ 28] = "hud/keycard";
item[ 29] = "hud/slayer_key";
item[ 30] = "hud/slayer_gate";
item[ 31] = "hud/mission_challenge_update";
item[ 32] = "hud/mission_challenge_complete";
item[ 33] = "hud/slayer_encounter";
item[ 34] = "hud/weapon_mastery_challenge_update";
item[ 35] = "hud/sentinel_battery_acquired";
item[ 36] = "hud/slayer_battery";
item[ 37] = "hud/automap";
item[ 38] = "hud/event/mission_complete";
item[ 39] = "hud/event/boss_complete";
item[ 40] = "hud/event/invasion_complete";
item[ 41] = "hud/event/master_level_complete";
item[ 42] = "hud/event/pvp_match_complete";
item[ 43] = "hud/rune_upgrade_token_acquired";
item[ 44] = "hud/bounty_spawned";
item[ 45] = "hud/secret_hint";
item[ 46] = "hud/checkpoint";
item[ 47] = "hud/weapon_mastery";
item[ 48] = "hud/hint";
item[ 49] = "hud/inventory_acquired";
item[ 50] = "hud/rad_suit_acquired";
item[ 51] = "hud/equipment_pickup_acquired";
item[ 52] = "hud/objective_show_last";
item[ 53] = "hud/milestone_complete";
```

```

        item[ 54] = "hud/season_level_up";
        item[ 55] = "hud/slayer_gate_e3";
        item[ 56] = "hud/weapon_points";
        item[ 57] = "hud/level_up";
        item[ 58] = "hud/secret_encounter_found";
        item[ 59] = "hud/milestone_complete_silent";
        item[ 60] = "hud/upgrade_reminder";
        item[ 61] = "hud/secret_encounter_failed";
        item[ 62] = "hud/secret_encounter_complete";
        item[ 63] = "hud/weapon_mastery_challenge_complete";
        item[ 64] = "hud/upgrade_reminder_praetor";
        item[ 65] = "hud/objective_pinned";
        item[ 66] = "hud/mission_challenge_intro";
        item[ 67] = "hud/gold_boss_start";
        item[ 68] = "hud/gold_boss_defeated";
        item[ 69] = "hud/extra_life_encounter_complete";
        item[ 70] = "hud/extra_life_encounter_failed";
        item[ 71] = "hud/extra_life_encounter_found";
        item[ 72] = "hud/master_level_start";
        item[ 73] = "hud/slayer_encounter_no_points";
        item[ 74] = "hud/extra_lifed_maxed";
        item[ 75] = "hud/extra_life_acquired_2";
        item[ 76] = "hud/extra_life_acquired_3";
        item[ 77] = "hud/dive_suit_acquired";
        item[ 78] = "hud/secret_encounter_dlc_complete";
        item[ 79] = "hud/secret_encounter_dlc_complete_skin_progress";
        item[ 80] = "hud/secret_encounter_dlc_complete_skin_unlocked";
        item[ 81] = "hud/support_rune_acquired";
        item[ 82] = "hud/player_cursed";
        item[ 83] = "hud/horde/timed_bounty_start";
        item[ 84] = "hud/horde/timed_bounty_completed";
        item[ 85] = "hud/horde/timed_bounty_failed";
        item[ 86] = "hud/horde/state_share_win";
        item[ 87] = "hud/horde/state_share_lose";
    }
}

WeaponManager = {
    masterAllWeapons = "campaign/master_all_weapons";
    requiredMasteredWeapons = {
        num = 8;
        item[ 0] = "campaign/master_ballista";
    }
}

```



```

        item[ 1 ] = "campaign/master_chaingun";
        item[ 2 ] = "campaign/master_combat_shotgun";
        item[ 3 ] = "campaign/master_heavy_cannon";
        item[ 4 ] = "campaign/master_plasma_rifle";
        item[ 5 ] = "campaign/master_rocket_launcher";
        item[ 6 ] = "campaign/master_super_shotgun";
    }

    dossierOrderWeaponList = {
        num = 9;
        item[ 0 ] = "weapon/player/shotgun";
        item[ 1 ] = "weapon/player/heavy_cannon";
        item[ 2 ] = "weapon/player/plasma_rifle";
        item[ 3 ] = "weapon/player/rocket_launcher";
        item[ 4 ] = "weapon/player/double_barrel";
        item[ 5 ] = "weapon/player/gauss_rifle";
        item[ 6 ] = "weapon/player/chaingun";
        item[ 7 ] = "weapon/player/bfg";
        item[ 8 ] = "weapon/player/unmaykr";
    }

}

runeManager = {
    masterRuneList = {
        num = 9;
        item[ 0 ] = "perk/player/runes/glory_kill_speed";
        item[ 1 ] = "perk/player/runes/glory_kill_dash";
        item[ 2 ] = "perk/player/runes/speed_boost_on_glory_kill";
        item[ 3 ] = "perk/player/runes/double_jump_air_control";
        item[ 4 ] = "perk/player/runes/modify_enemy_stagger_duration";
        item[ 5 ] = "perk/player/runes/activate_focus_on_death_blow";
        item[ 6 ] = "perk/player/runes/target_strike";
        item[ 7 ] = "perk/player/runes/decrease_equipment_recharge";
        item[ 8 ] = "perk/player/runes/blood_punch_loot_on_damage";
    }

    runePerkGroup = "perkgroups/runes";
    supportRunePerkGroup = "perkgroups/support_runes";
    runeSlotReq = {
        ptr = {
            ptr[ 0 ] = 1;
            ptr[ 1 ] = 2;
            ptr[ 2 ] = 3;
        }
    }
}

```

```

    {}
    {}
    {}$suitManager = {
        {}$masterSuitGroup = "perkgroups/suit";
        {}$fundamentals = {
            {}$display = "#str_praetor_fundamentals";
            {}$description = "#str_praetor_fundamentals_desc";
            {}$iconMaterial = "art/ui/dossier/icons/preator_suit/category/fundamentals";
            {}$suitUpgrades = {
                {}$hum = 4;
                {}$item[0] = "perk/player/suit/fundamentals/ledge_grab_speed";
                {}$item[1] = "perk/player/suit/fundamentals/weapon_change_speed";
                {}$item[2] = "perk/player/suit/dash/dash_regen_delay";
                {}$item[3] = "perk/player/suit/dash/dash_gk_restore_dash";
            {}$}
        {}$}
        {}$extermination = {
            {}$index = 1;
            {}$display = "#str_praetor_extermination";
            {}$description = "#str_praetor_extermination_desc";
            {}$iconMaterial = "art/ui/dossier/icons/preator_suit/category/extermination";
            {}$suitUpgrades = {
                {}$hum = 4;
                {}$item[0] = "perk/player/equipment/frag_reduce_cooldown";
                {}$item[1] = "perk/player/equipment/frag_concussive_blast";
                {}$item[2] = "perk/player/equipment/frag_max_capacity";
                {}$item[3] = "perk/player/equipment/frag_cluster_bombs";
            {}$}
        {}$}
        {}$selfPreservation = {
            {}$index = 2;
            {}$display = "#str_praetor_self_preservation";
            {}$description = "#str_praetor_self_preservation_desc";
            {}$iconMaterial = "art/ui/dossier/icons/preator_suit/category/self_preservation";
            {}$suitUpgrades = {
                {}$hum = 4;
                {}$item[0] = "perk/player/suit/self_preservation/reduce_hazard_damage";
                {}$item[1] = "perk/player/suit/self_preservation/reduce_self_weapon_damage";
                {}$item[2] = "perk/player/suit/extermination/barrels_respawn";
                {}$item[3] = "perk/player/suit/extermination/barrels_drop_ammo";
            {}$}
        {}$}
    {}$}

```

```

    {}
    {}powerups = {
    {}index = 3;
    {}display = "#str_praetor_powerups";
    {}description = "#str_praetor_powerups_desc";
    {}iconMaterial = "art/ui/dossier/icons/preator_suit/category/powerups";
    {}suitUpgrades = {
    {}hum = 4;
    {}item[0] = "perk/player/equipment/ice_reduce_cooldown";
    {}item[1] = "perk/player/equipment/ice_extend_duration";
    {}item[2] = "perk/player/equipment/ice_health_drops";
    {}item[3] = "perk/player/equipment/ice_melee_shatter";
    {}
    {}
    {}exploration = {
    {}index = 4;
    {}display = "#str_praetor_exploration";
    {}description = "#str_praetor_exploration_desc";
    {}iconMaterial = "art/ui/dossier/icons/preator_suit/category/navigation";
    {}suitUpgrades = {
    {}hum = 4;
    {}item[0] = "perk/player/suit/exploration/automap_shows_map_station";
    {}item[1] = "perk/player/suit/exploration/automap_shows_progression_items";
    {}item[2] = "perk/player/suit/exploration/automap_increased_fog_radius";
    {}item[3] = "perk/player/suit/exploration/dossier_shows_progression_items";
    {}
    {}
    {}dash = {
    {}index = 5;
    {}display = "#str_praetor_dash";
    {}description = "#str_praetor_dash_desc";
    {}iconMaterial = "art/ui/dossier/icons/preator_suit/category/dash";
    {}suitUpgrades = {
    {}hum = 4;
    {}item[0] = "perk/player/suit/self_preservation/reduce_hazard_damage";
    {}item[1] = "perk/player/suit/self_preservation/reduce_self_weapon_damage";
    {}item[2] = "perk/player/suit/extermination/barrels_respawn";
    {}item[3] = "perk/player/suit/extermination/barrels_drop_ammo";
    {}
    {}
    {}argentCellPerkGroup = "perkgroups/argent_cell";

```

```

    [][suitPerkGroup = "perkgroups/suit";
    [][cheatcode = "CHEAT_CODE_TYPE_QUAKECON_MODE";
    [][allAcqTutorial = "tutorialevent/ingame_tutorial/quakecon_mode";
    []}
    [][unlockableManager = {
    [][missionChallenges = "missionchallenge/main";
    [][milestones = "milestonelist/main";
    []}
    [][statManager = {
    [][explorationHintOptions = {
    [][hum = 3;
    [][item[ 0] = {
    [][requiredRules = {
    [][hum = 1;
    [][item[ 0] = {
    [][stat = "STAT_CHALLENGES_COMPLETED";
    [][value = 0;
    [][operation = "RULE_OPERATION_EQUALS";
    [][}
    [][}
    [][resultDataDecl = "hint/gameplay/mission_challenge";
    [][}
    [][item[ 1] = {
    [][requiredRules = {
    [][hum = 1;
    [][item[ 0] = {
    [][stat = "STAT_SECRETS_FOUND";
    [][secondStat = "STAT_SECRETS";
    [][secondStatDuration = "DUR_LEVEL";
    [][value = 50;
    [][operation = "RULE_OPERATION_LESS_THAN_PERCENTAGE";
    [][}
    [][}
    [][bonusRules = {
    [][hum = 1;
    [][item[ 0] = {
    [][stat = "STAT_SECRETS_FOUND";
    [][secondStat = "STAT_SECRETS";
    [][secondStatDuration = "DUR_LEVEL";
    [][value = 25;
    [][operation = "RULE_OPERATION_LESS_THAN_PERCENTAGE";

```

```

        }
    }
    resultDataDecl = "hint/exploration/secrets";
}

item[ 2] = {
    requiredRules = {
        hum = 1;
        item[ 0] = {
            stat = "STAT_SLAYER_GATE_USED";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_EQUALS";
        }
    }
    resultDataDecl = "hint/exploration/slayer_gate";
}

combatHintOptions = {
    hum = 25;
    item[ 0] = {
        requiredRules = {
            hum = 1;
            item[ 0] = {
                stat = "STAT_REVENANT_KILLED";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
        resultDataDecl = "hint/combat/revenant";
    }
    item[ 1] = {
        requiredRules = {
            hum = 1;
            item[ 0] = {
                stat = "STAT_HELL_MARINE_KILLED";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
    }
}

```

```

        resultDataDecl = "hint/combat/soldier_blasters";
    }

    item[ 2 ] = {
        requiredRules = {
            hum = 1;
            item[ 0 ] = {
                stat = "STAT_PINKY_KILLED";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
    }

    resultDataDecl = "hint/combat/pinky";
    }

    item[ 3 ] = {
        requiredRules = {
            hum = 1;
            item[ 0 ] = {
                stat = "STAT_BARON_KILLED";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
    }

    resultDataDecl = "hint/combat/baron";
    }

    item[ 4 ] = {
        requiredRules = {
            hum = 1;
            item[ 0 ] = {
                stat = "STAT_CACODEMON_KILLED";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
    }

    resultDataDecl = "hint/combat/cacodemon";
    }

    item[ 5 ] = {
        requiredRules = {
            hum = 1;

```

```

        item[ 0] = {
            stat = "STAT_MANCUBUS_KILLED";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }

    resultDataDecl = "hint/combat/mancubus_fire";
}

item[ 6] = {
    requiredRules = {
        num = 1;
        item[ 0] = {
            stat = "STAT_LOST_SOUL_KILLED";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }

    resultDataDecl = "hint/combat/lost_soul";
}

item[ 7] = {
    requiredRules = {
        num = 1;
        item[ 0] = {
            stat = "STAT_HELL_SHOTGUNNER_KILLED";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }

    resultDataDecl = "hint/combat/soldier_shield";
}

item[ 8] = {
    requiredRules = {
        num = 1;
        item[ 0] = {
            stat = "STAT_WEAPON_SHOTGUN_DAMAGE";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }
}

```

```

    }
    }
    resultDataDecl = "hint/combat/shotgun";
    }
    item[ 9] = {
        requiredRules = {
            num = 1;
            item[ 0] = {
                stat = "STAT_WEAPON_DBL_SHOTGUN_DAMAGE";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
        resultDataDecl = "hint/combat/double_barrel";
    }
    item[ 10] = {
        requiredRules = {
            num = 1;
            item[ 0] = {
                stat = "STAT_WEAPON_ASS_RIFLE_DAMAGE";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
        resultDataDecl = "hint/combat/heavy_cannon";
    }
    item[ 11] = {
        requiredRules = {
            num = 1;
            item[ 0] = {
                stat = "STAT_WEAPON_PLASM_RIFLE_DAMAGE";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
        resultDataDecl = "hint/combat/plasma_rifle";
    }
    item[ 12] = {

```



```

        requiredRules = {
            hum = 1;
            item[ 0] = {
                stat = "STAT_WEAPON_ROCKET_LAUNCHER_KILLS";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
        resultDataDecl = "hint/combat/rocket_launcher";
    }
    item[ 13] = {
        requiredRules = {
            hum = 1;
            item[ 0] = {
                stat = "STAT_WEAPON_CHAINGUN_KILLS";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
        resultDataDecl = "hint/combat/chaingun";
    }
    item[ 14] = {
        requiredRules = {
            hum = 1;
            item[ 0] = {
                stat = "STAT_WEAPON_GAUSS_KILLS";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
        resultDataDecl = "hint/combat/gauss_rifle";
    }
    item[ 15] = {
        requiredRules = {
            hum = 1;
            item[ 0] = {
                stat = "STAT_WEAPON_BFG_KILLS";
                statDuration = "DUR_GAME";
            }
        }
    }

```

```

        value = 0;
        operation = "RULE_OPERATION_GREATER_THAN_VALUE";
    }
}

resultDataDecl = "hint/combat/bfg";
}

item[16] = {
    requiredRules = {
        num = 1;
        item[0] = {
            stat = "STAT_WEAPON_CHAINSAW_KILLS";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }
    resultDataDecl = "hint/combat/chainsaw";
}

item[17] = {
    requiredRules = {
        num = 1;
        item[0] = {
            stat = "STAT_SHOTGUN_FULL_AUTO_EARNED";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }
    resultDataDecl = "hint/combat/shotgun_full_auto";
}

item[18] = {
    requiredRules = {
        num = 1;
        item[0] = {
            stat = "STAT_SHOTGUN_STICKY_BOMB_EARNED";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }
    resultDataDecl = "hint/combat/shotgun_pop_rockets";
}

```

```

    {}
    item[ 19] = {
        requiredRules = {
            hum = 1;
            item[ 0] = {
                stat = "STAT_HAR_SCOPE_EARNED";
                statDuration = "DUR_GAME";
                value = 0;
                operation = "RULE_OPERATION_GREATER_THAN_VALUE";
            }
        }
    }
    resultDataDecl = "hint/combat/heavy_cannon_bolt_action";
}
item[ 20] = {
    requiredRules = {
        hum = 1;
        item[ 0] = {
            stat = "STAT_HAR_MICROMISSILE_EARNED";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }
    resultDataDecl = "hint/combat/heavy_cannon_burst_detonate";
}
item[ 21] = {
    requiredRules = {
        hum = 1;
        item[ 0] = {
            stat = "STAT_CHAINGUN_TURRET_EARNED";
            statDuration = "DUR_GAME";
            value = 0;
            operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        }
    }
    resultDataDecl = "hint/combat/chaingun_turret";
}
item[ 22] = {
    requiredRules = {
        hum = 1;
        item[ 0] = {

```

```

        [ ]stat = "STAT_PLASMA_HEATWAVE_EARNED";
        [ ]statDuration = "DUR_GAME";
        [ ]value = 0;
        [ ]operation = "RULE_OPERATION_GREATER_THAN_VALUE";
        [ ]
        [ ]
        [ ]resultDataDecl = "hint/combat/plasma_rifle_aoe";
        [ ]
        [ ]item[ 23] = {
            [ ]requiredRules = {
                [ ]hum = 1;
                [ ]item[ 0] = {
                    [ ]stat = "STAT_ROCKET_LAUNCHER_DETONATE_EARNED";
                    [ ]statDuration = "DUR_GAME";
                    [ ]value = 0;
                    [ ]operation = "RULE_OPERATION_GREATER_THAN_VALUE";
                    [ ]
                    [ ]
                    [ ]resultDataDecl = "hint/combat/rocket_launcher_detonate";
                    [ ]
                    [ ]item[ 24] = {
                        [ ]requiredRules = {
                            [ ]hum = 1;
                            [ ]item[ 0] = {
                                [ ]stat = "STAT_ROCKET_LAUNCHER_LOCKON_EARNED";
                                [ ]statDuration = "DUR_GAME";
                                [ ]value = 0;
                                [ ]operation = "RULE_OPERATION_GREATER_THAN_VALUE";
                                [ ]
                                [ ]
                                [ ]resultDataDecl = "hint/combat/rocket_launcher_lock_mod";
                                [ ]
                                [ ]
                                [ ]
                                [ ]collectibleManager = {
                                    [ ]collectibleGroupUIData = "collectiblecategorydata";
                                    [ ]
                                    [ ]uiWalkthroughManager = {
                                        [ ]walkthroughTutorials = {
                                            [ ]praetorSuitTutorial = "uiwalkthrough_praetorsuit";
                                            [ ]modBotTutorial = "uiwalkthrough_modbot";

```

```

    [argmentCellTutorial = "uiwalkthrough_argentcell";
    [runeTutorial = "uiwalkthrough_runes";
    [runeTutorialInteractState = "used_rune";
    [supportRuneTutorial = "uiwalkthrough_supportrunes";
    }
    }
    [soundPackManager = {
    [soundPacks = {
    [hum = 1;
    [item[ 0] = "classic_doomguy_sounds";
    }
    }
    [scoreManager = {
    [rubric = "horde";
    }
    [violenceKillEvents = {
    [hum = 18;
    [item[ 0] = "violenceevent/weapon_mastery/shotgun/full_auto_kill_pinky";
    [item[ 1] = "violenceevent/weapon_mastery/rocket_launcher/lock_on_kill_prowler";
    [item[ 2] = "violenceevent/weapon_mastery/heavy_cannon/piercing_bolt_headshot";
    [item[ 3] = "violenceevent/weapon_mastery/ballista/arbalest_kill_cacodemons";
    [item[ 4] = "violenceevent/weapon_mastery/ballista/destroyer_kill_n_in_one_shot";
    [item[ 5] = "violenceevent/weapon_mastery/chaingun/mobile_turret_kill_n_in_one_deployment";
    [item[ 6] = "violenceevent/weapon_mastery/plasma_rifle/heat_blast_kill_n_in_one_shot";
    [item[ 7] = "violenceevent/mission_challenge/bfg_kill_doom_hunter";
    [item[ 8] = "violenceevent/mission_challenge/chapel_of_ghouls";
    [item[ 9] = "violenceevent/mission_challenge/balls_to_the_wall";
    [item[ 10] = "violenceevent/mission_challenge/master_of_puppets";
    [item[ 11] = "violenceevent/weapon_mastery/plasma_rifle/microwave_kill_n_in_one_shot";
    [item[ 12] = "violenceevent/mission_challenge/bonded_by_blood";
    [item[ 13] = "violenceevent/mission_challenge/external_combustion";
    [item[ 14] = "violenceevent/mission_challenge/hammer_smashed_face";
    [item[ 15] = "violenceevent/mission_challenge/kill_n_demons_with_one_frag";
    [item[ 16] = "violenceevent/mission_challenge/angel_of_death";
    [item[ 17] = "violenceevent/pvp/timeout";
    }
    [violenceHitEvents = {
    [hum = 8;
    [item[ 0] = "violenceevent/weapon_mastery/heavy_cannon/micro_missile_kill_n_demons";
    [item[ 1] = "violenceevent/mission_challenge/through_the_fire_and_flames";
    [item[ 2] = "violenceevent/dash_into_pinky";

```

```

[[[item[3] = "violenceevent/pvp/fire_starter";
[[[item[4] = "violenceevent/pvp/ignition";
[[[item[5] = "violenceevent/pvp/wolf_master";
[[[item[6] = "violenceevent/pvp/dodge_this";
[[[item[7] = "violenceevent/pvp/punch_drunk";
[[[
[[[
[[playerProps = "player/default";
[[playerMechanicLedgeGrab = {
[[[animAliases = {
[[[ledgePullUp = "ledgeGrab/ledgePullUp";
[[[ledgePullUpMantle = "ledgeGrab/ledgePullUpMantle";
[[[ledgePullUpRounded = "ledgeGrab/ledgePullUpRounded";
[[[ledgePullUpMantleRounded = "ledgeGrab/ledgePullUpMantleRounded";
[[[ledgePullUpAngled = "ledgeGrab/ledgePullUpAngled";
[[[ledgePullUpMantleAngled = "ledgeGrab/ledgePullUpMantleAngled";
[[[ledgeClimbUp = "ledgeGrab/ledgeClimbUp";
[[[ledgeClimbUpMantle = "ledgeGrab/ledgeClimbUpMantle";
[[[railingPullUp = "ledgeGrab/railingPullUp";
[[[railingPullUpMantle = "ledgeGrab/railingPullUpMantle";
[[[
[[[thirdPersonAnimIndex = {
[[[ledgePullUpMantle = 1;
[[[ledgePullUpMantleRounded = 1;
[[[ledgePullUpMantleAngled = 1;
[[[ledgeClimbUp = 4;
[[[ledgeClimbUpMantle = 5;
[[[railingPullUp = 2;
[[[railingPullUpMantle = 3;
[[[
[[[
[[playerMechanicWallGrab = {
[[[initiateMaxAngle = 115;
[[[maxDeltaYaw = 115;
[[[constrainedViewAnglesRate = 250;
[[[speedNormal = 12.3820009;
[[[speedSprint = 12.3820009;
[[[speedDismountPerp = 5.71500015;
[[[speedDismountUp = 5.90500021;
[[[speedDismountFwd = 9.52500057;
[[[sound_mount = "player/foley/wallgrab/player_foley_wallgrab_mount";

```

```

    sound_move = "player/foley/wallgrab/player_foley_wallgrab_move";
    sound_dismount = "player/foley/wallgrab/player_foley_wallgrab_dismount";
}

playerMechanicWallClimb = {
    anims = {
        climb_up_cycle_right = "wall_climb/climb_up_cycle_right";
        climb_up_cycle_left = "wall_climb/climb_up_cycle_left";
        climb_down_cycle_right = "wall_climb/climb_down_cycle_right";
        climb_down_cycle_left = "wall_climb/climb_down_cycle_left";
        climb_right_cycle = "wall_climb/climb_right_cycle";
        climb_left_cycle = "wall_climb/climb_left_cycle";
        climb_into_up_right = "wall_climb/climb_up_into_right";
        climb_into_up_left = "wall_climb/climb_up_into_left";
        climb_into_down_right = "wall_climb/climb_down_into_right";
        climb_into_down_left = "wall_climb/climb_down_into_left";
        climb_into_right = "wall_climb/climb_right_into";
        climb_into_left = "wall_climb/climb_left_into";
        climb_out_up_right = "wall_climb/climb_up_out_right";
        climb_out_up_left = "wall_climb/climb_up_out_left";
        climb_out_down_right = "wall_climb/climb_down_out_right";
        climb_out_down_left = "wall_climb/climb_down_out_left";
        climb_out_right = "wall_climb/climb_right_out";
        climb_out_left = "wall_climb/climb_left_out";
        climb_out_right_mid = "wall_climb/climb_left_out_mid";
        climb_out_left_mid = "wall_climb/climb_right_out_mid";
        idle = "wall_climb/idle";
        to_pointing = "wall_climb/climb_to_pointing";
        from_pointing = "wall_climb/climb_from_pointing";
        mount_air = "wall_climb/mount_air";
        mount_ground = "wall_climb/mount_ground";
        look_left = "wall_climb/look_left";
        look_left_up_45 = "wall_climb/look_left_up_45";
        look_left_up_90 = "wall_climb/look_left_up_90";
        look_right = "wall_climb/look_right";
        look_right_up_45 = "wall_climb/look_right_up_45";
        look_right_up_90 = "wall_climb/look_right_up_90";
        look_right_idle = "wall_climb/look_right_idle";
        look_left_idle = "wall_climb/look_left_idle";
    }
    globalTraceDistance = 10;
    viewPitchTable = "wall_climb_view_pitch";
}

```

```

    verticalMoveTable = "wall_climb_slowmove";
    horizontalMoveTable = "wall_climb_horizontalmove";
    attachToWallSoundDecl = "play_climb_hit";
    jumpOffWallSoundDecl = "play_climb_off";
    removeHandFromWallSoundDecl = "play_climb_hand";
    attachHandToWallSoundDecl = "play_climb_hand_wall";
    distanceToWallToActivate = 7.5;
    startMoveDelaySec = 0;
    slowMoveDelayBetweenSec = 0;
    initialMovementDelaySec = 0.300000012;
    maxDeltaYawViewAngle = 220;
    goalSpeedVerticalAccelRate = 3;
    goalSpeedHorizontalAccelRate = 5;
    goalSpeedVerticalJumpAccelRate = 5;
    goalSpeedHorizontalJumpAccelRate = 5;
    goalSpeedVerticalDecelRate = 5;
    goalSpeedHorizontalDecelRate = 5;
    jumpToWallSpeed = 40;
    overdriveDecl = "statuseffect/overdrive";
    wallClimbText = "#str_climb_wall";
}

playerMechanicGorillaBarJump = {
    swingAnimation = "monkeyBar/grab";
    maxLookAngleFromBar = 90;
    screenShakeDecl = "advancedscreenviewshake/monkey_bar";
}

playerMechanicInteract = {
    animData = {
        interactAnimDatas = {
            hum = 3;
            item[ 0 ] = {
                type = "PLAYER_MECHANIC_THIRD_PERSON_INTERACT_TYPE_GENERIC";
            }
            item[ 1 ] = {
                type = "PLAYER_MECHANIC_THIRD_PERSON_INTERACT_TYPE_SENTINEL_CRYSTAL";
                animIndex = 1;
            }
            item[ 2 ] = {
                type = "PLAYER_MECHANIC_THIRD_PERSON_INTERACT_TYPE_BUTTON";
                animIndex = 2;
                putroAnimIndex = 1;
            }
        }
    }
}

```



```

    {}
    {}
    {}
    {}statusEffect = "statuseffect/invasion/slayer/interact_invulnerability";
    {}
    {}playerAbilityFuelRate = 3;
    {}playerHealth = {
        {}components = {
            {}components[0] = {
                {}regenInterval = {
                    {}value = 0.02999999993;
                }
            }
            {}regenDelay = {
                {}value = 12;
            }
        }
        {}components[1] = {
            {}max = {
                {}defaultValue = 50;
            }
            {}drainLimit = 50;
            {}starting = {
                {}defaultValue = 0;
            }
            {}regenInterval = {
                {}value = 0.02999999993;
            }
            {}regenDelay = {
                {}value = 12;
            }
        }
    }
    {}energyShellDecl = "pickup/energy_shell";
    {}energyShellDeclMP = "pickup/energy_shell_mp";
    {}playerHealthRTPC = "player_health";
    {}soundResetStates = {
        {}num = 3;
        {}item[0] = "swimming/none";
        {}item[1] = "cinematic/none";
        {}item[2] = "curse/none";
    }

```

```

    }
    goreComponent = {
        goreContainer = "player";
    }
    damageScaleTableComponent = {
        list = {
            num = 7;
            item[0] = {
                declDamage = "damage/special/ability_dash";
                declTable = "player/damage_volume_test";
            }
            item[1] = {
                declDamage = "damage/hazard/bio";
                declTable = "player/bio_hazard_damage";
            }
            item[2] = {
                declDamage = "damage/hazard/lava";
                declTable = "player/lava_hazard_damage";
            }
            item[3] = {
                declDamage = "damage/hazard/slime";
                declTable = "player/slime_hazard_damage";
                resetTableOnExitTimeMS = 2000;
            }
            item[4] = {
                declDamage = "damage/hazard/water_toxic";
                declTable = "player/slime_hazard_damage";
                resetTableOnExitTimeMS = 2000;
            }
            item[5] = {
                declDamage = "damage/hazard/slime_sac";
                declTable = "player/slime_sac_hazard_damage";
                resetTableOnExitTimeMS = 2000;
            }
            item[6] = {
                declDamage = "damage/hazard/poison_bog";
                declTable = "player/poison_bog_hazard_damage";
                resetTableOnExitTimeMS = 2000;
            }
        }
    }
}

```

```

playerHud = {
    declHUD = "default";
}

playerDemoHud = {
    declHUD = "demo";
}

demonPlayerHud = {
    declHUD = "demon";
}

bodyAnimWeb = "animweb/player/doom_marine_3p";
suffocationDamage = "damage/base/suffocation";
waterBoostScaleMax = 4.75;
waterBoostSound = "play_swish_frantic";
playerRevive = "play_player_revive";
doubleJumpSound = "play_double_jump";
audioLogComponent = {
    stories = {
        hum = 1;
        item[0] = "dev_story/attack_of_the_jeff";
    }
}

screenFXDecl = "player/screen_damage";
playerAimEventDecl = "base/misc/playeraim";
playerInteractionEventDecl = "base/misc/playerinteraction";
bodyEventDecl = "base/misc/body";
thirdPersonFXDecl = "player/tp_body";
combatEncounterScoringDecl = "base";
nightmareMarkerDef = "interact/nightmaremarker/public";
nightmareMarkerDefFriends = "interact/nightmaremarker/public";
nightmareMarkerDefBestRun = "interact/nightmaremarker/furthest";
nightmareMarkerDefPublic = "interact/nightmaremarker/public";
maxNumMarkersLvl = 30;
gamePauseSound = "play_game_pause";
gameUnpauseSound = "play_game_resume";
playerInvasionData = "default";
weaponFists = "weapon/player/fists";
weaponFistsBerserk = "weapon/player/fists_berserk";
weaponFistsDoom5Melee = "weapon/player/fists_doom5melee";
weaponFistsSwim = "weapon/player/fists_swim";
weaponPointingArm = "weapon/player/pointing_arm";
weaponCrucible = "weapon/player/crucible";

```

```

    equipmentLauncherLeftDecl = "equipmentlauncher/equipmentlauncherleft";
    equipmentLauncherRightDecl = "equipmentlauncher/equipmentlaucherright";
    weaponFlameBelchLeft = "weapon/player/equipment_flame_belch";
    weaponFlameBelchRight = "weapon/player/equipment_flame_belch_right";
    weaponChainsaw = "weapon/player/chainsaw";
    weaponPlasma = "weapon/player/plasma_rifle";
    weaponChaingun = "weapon/player/chaingun";
    weaponHammer = "weapon/player/hammer";
    guiInfo = {
        icon = "art/ui/pvp_playericons/ico_pvp_players_doomguy_hud";
        iconAlt = "art/ui/pvp_playericons/ico_pvp_players_doomguy_hud_dead";
    }
    lockonData = {
        lockon_acquiringSound = "play_revenant_lock_warning";
        lockon_lockedSound = "play_demonplayer_missileslocked";
        lockon_broken = "play_rocket_lockon_break_alert";
    }
    friendlyRepulsorRadius = 0.25;
    enemyRepulsorRadius = 0.25;
    controlRespawnedFromDeath = true;
    canRespawn = true;
}

```

Revision #3

Created 22 August 2023 20:17:05 by Velser

Updated 12 February 2024 20:59:52 by Velser