
Subject: WeaponBag

Posted by [Neijwiert](#) on Sat, 30 Nov 2013 17:29:39 GMT

[View Forum Message](#) <> [Reply to Message](#)

FIXED IT

There seems to be something wrong with the WeaponBagClass.

I have a situation where I select the most optimal weapon for a bot against a certain object. Everytime I have the exact same set-up, yet on some occasions the WeaponBagClass::Get_Count() returns 0. However I can see the weapon on the actual soldier in-game and it's the exact same soldier preset and everything I had before.

I do call my function multiple times, so it's not like those usual problems that you have to wait x time after creation to do y.

I can provide my whole code, so you guys can recreate this problem if you can't identify it in my code.

Toggle Spoiler

```
WeaponClass *NTC_AI::Select_Effective_Weapon(GameObject *Target, int &Status)
```

```
{  
    if(!Target)  
    {  
        Status = NO_WEAPON_FOUND;  
        return 0;  
    }  
}
```

```
DamageableGameObj *DTarget = Target->As_DamageableGameObj();
```

```
if(!DTarget)  
{  
    Status = NO_WEAPON_FOUND;  
    return 0;  
}
```

```
DefenseObjectClass *DefTarget = DTarget->Get_Defense_Object();
```

```
if(!DefTarget)  
{  
    Status = NO_WEAPON_FOUND;  
    return 0;  
}
```

```
PhysicalGameObj *PObj = Owner()->As_PhysicalGameObj();
```

```
if(!PObj)  
{  
    Status = NO_WEAPON_FOUND;  
    return 0;  
}
```

```

ArmedGameObj *ArmedObj = PObj->As_ArmedGameObj();
if(!ArmedObj)
{
    Status = NO_WEAPON_FOUND;
    return 0;
}

WeaponBagClass *Bag = ArmedObj->Get_Weapon_Bag();
if(!Bag)
{
    Status = NO_WEAPON_FOUND;
    return 0;
}

unsigned int ArmorType = DefTarget->Get_Skin();
if(DefTarget->Get_Shield_Strength() > DefTarget->Get_Health())
    ArmorType = (unsigned int)DefTarget->Get_Shield_Type();

float HighestDamage = 0;
int WeaponIndex = -1;
bool WasSecondary = false;
float HighestWeaponRange = 0;

float Distance = Commands->Get_Distance(Commands->Get_Position(Owner()),
Commands->Get_Position(Target));

int WeaponCount = Bag->Get_Count();
for(int x = 0; x < WeaponCount; x++)
{
    Console_Output("looping trough a weapon\n");
    WeaponClass *Weapon = Bag->Peek_Weapon(x);
    if(Weapon && Weapon->Get_Clip_Rounds() + Weapon->Get_Inventory_Rounds() > 0)
    {
        Console_Output("Got a weapon and ammo\n");
        AmmoDefinitionClass *PrimaryAmmo =(AmmoDefinitionClass
*)DefinitionMgrClass::Find_Definition(Weapon->Get_Definition()->PrimaryAmmoDefID, false);
        AmmoDefinitionClass *SecondaryAmmo = (AmmoDefinitionClass
*)DefinitionMgrClass::Find_Definition(Weapon->Get_Definition()->SecondaryAmmoDefID, false);

        float CurrentHighestDamage = 0;
        bool Secondary = false;

        if(PrimaryAmmo)
        {
            if(PrimaryAmmo->Range >= Distance)
                CurrentHighestDamage = ArmorWarheadManager::Get_Damage_Multiplier(ArmorType,
PrimaryAmmo->Warhead) * PrimaryAmmo->Damage;

```

```

if(PrimaryAmmo->Range > HighestWeaponRange)
    HighestWeaponRange = PrimaryAmmo->Range;
}

if(SecondaryAmmo)
{
    if(SecondaryAmmo->Range >= Distance)
    {
        float SecondaryDamage = ArmorWarheadManager::Get_Damage_Multiplier(ArmorType,
SecondaryAmmo->Warhead) * SecondaryAmmo->Damage;
        if(SecondaryDamage > CurrentHighestDamage)
        {
            Secondary = true;
            CurrentHighestDamage = SecondaryDamage;
        }
    }
}

if(SecondaryAmmo->Range > HighestWeaponRange)
    HighestWeaponRange = SecondaryAmmo->Range;
}

if(CurrentHighestDamage > HighestDamage)
{
    HighestDamage = CurrentHighestDamage;
    WeaponIndex = x;
    WasSecondary = Secondary;
}
}
}

if(WeaponIndex != -1)
{
    WeaponClass *Weapon = Bag->Peek_Weapon(WeaponIndex);
    if(Weapon)
    {
        if(WasSecondary)
            Status = WEAPON_SECONDARY;
        else
            Status = WEAPON_PRIMARY;

        Console_Output("A\n");
        return Weapon;
    }
    else
    {
        Console_Output("B\n");
        Status = NO_WEAPON_FOUND;
    }
}

```

```
    return 0;
}
}
else
{
if(HighestWeaponRange < Distance)
{
    Console_Output("C\n");
    Status = OUT_OF_WEAPON_RANGE;
    return 0;
}
else
{
    Console_Output("D\n");
    Status = NO_WEAPON_FOUND;
    return 0;
}
}
}
```