

Slaved Behaviour

Heute möchte ich euch ein relativ simples System aus Edain näher erklären, das sich mit Slaved-Objecten befasst. Hierbei beziehe ich mich auf das System der Creeps, die ein bestimmtes Gebäude automatisch verteidigen. Es gibt verschiedene Möglichkeiten dieses System umzusetzen, so zum Beispiel auch das System der Lorien-Bogenschützen, die sich per Achsenverschiebung an einem bestimmten Punkt befinden und von selbst attackieren, ohne sich dabei aber zu bewegen. Mit diesen Funktionen kann man auch tolle Ambiente-Erscheinungen entstehen lassen, sprich Tiere (ohne spezielle Animationen) an ein Gebäude heften.

Für dieses Tutorial setze ich eine gewisse Grundkenntnis im Bereich des Codings voraus.

Folgende inis benötigt ihr:

Gebäude.ini
Einheiten.ini

Ich kenne mittlerweile 3 verschiedene Systeme, beginnen möchte ich mit dem Standard-Creep-System.

1) Geht in eine bestimmte Gebäude.ini, die diese Funktion erhalten soll. Fügt dort folgenden Code ein:

```
Behavior = SpawnBehavior ModuleTag_SpawnFighters
    TriggeredBy             = Upgrade_StructureLevel1;Upgrade-Voraussetzung
    SpawnNumber             = 4 ;Anzahl der Einheiten
    InitialBurst            = 4
    SpawnTemplateName      = Grundobject_Slaved ;Object
    SpawnReplaceDelay      = 20000 ;Intervalle, bevor neue gespawnt werden
    CanReclaimOrphans      = Yes
End
```

Nun geht ihr in die ini der Einheit und erstellt ein Childobject, welches diese Codes beinhalten sollte:

(name nur Testweise)

```
ChildObject Grundobject_Slaved Grundobject
```

```
    CommandPoints = 0

    EvaEnemyObjectSightedEvent = GoblinCreepSighted ; Special event for
goblins showing up. Helps distinguish the real enemies from random creeps
    EvaEnemyObjectSightedAfterRespawnEvent = GoblinCreepSighted ; Event to run
after the Goblin Lair is respawned. WARNING: This INI field only works for
"SlavedUpdate.UseSlaverAsControlForEvaObjectSightedEvents = Yes" units

    ; Should be the same as MordorGoblinSwordsman but with CREEP
    KindOf = +CREEP

    VisionRange = CREEP_VISION

    ThreatBreakdown MordorGoblinSwordsmanSlaved_ThreatBreakdown
        AIKindOf = CREEP
End
```

```

    Behavior = SlavedUpdate ModuleTag_Slave
        GuardMaxRange      = 250 ;Wie weit sich die Einheit vom Gebäude
entfernen darf
        GuardWanderRange   = 40  ;Wie lange sich die Einheit im Intervall
bewegt
        UseSlaverAsControlForEvaObjectSightedEvents = Yes
    End
End

```

Ihr könnt jetzt weitere Units per Upgrade an das Gebäude heften, indem ihr weitere Slaved-Behaviour (Spawnbehaviour) an das Gebäude knüpft.

2) Die zweite Variante ist schon um einiges komplizierter und von mir selbst entwickelt worden, orientiert am System des Trebuchet. Wichtig hierbei ist, dass ihr mit Renx umgehen könnt bzw. dieses zumindest benutzen könnt.

Geht in eure Gebäude.ini und fügt folgendes ein:

```

    Behavior = GrantUpgradeCreate ModuleTag_ActLikeAnOldUpgrade
        UpgradeToGrant      = Upgrade_TrebuchetTurret
        GiveOnBuildComplete = Yes
    End

    Behavior = ObjectCreationUpgrade MakeTheFreeTreb1
        TriggeredBy         = Upgrade_TrebuchetTurret
        Delay                = 0.0
        ThingToSpawn        = ElvenLorienArcher_SlavedForBuildings
        Offset               = X:-3 Y:14 Z:84
        FadeInTime          = 100
    End

```

Sobald das Gebäude gebaut wurde, erhält es ein Upgrade, welches eine Unit an einem speziellen Punkt spawnnt. Dieser Punkt ist dieser hier: **x:-3 y:14 z:84**

Wir befinden uns in einem dreidimensionalen Raum, das heißt wir müssen alle Achsen benutzen und das Object genau an einer Stelle absetzen, an der es uns auch logisch erscheint. Sucht euch also einen Punkt an eurem Gebäude wo eurer Meinung nach eine Units als Verteidigung stehen sollte. Geht dafür in Renx, importiert euer Gebäude, und lest eure Werte an der unteren Achsenbeschreibung ab. Dafür fahrt ihr langsam mit eure Maus über einem Punkt und kopiert die Werte in die ini unter Offset.

Ihr könnt dafür beliebig viele ObjectCreation-Objecte verwenden, achtet aber darauf, immer andere Punkte zu nehmen.

Damit auch alles reibungslos abläuft, müssen wir jetzt einige Dinge beachten. Zum Einen muss das Object natürlich auch sterben, wenn das Gebäude stirbt und es muss sofort im Boden verschwinden,

Ich habe euch jetzt einfach mal die nötigen Codes für einen Lorien-Bogenschützen zurecht gelegt. Schaut es euch an und kopiert die wichtigen Einträge für euch:

```

ChildObject ElvenLorienArcher_SlavedForBuildings ElvenLorienArcher

    VisionSide          = 100%
    VisionRear          = 100%
    ShroudClearingRange = 300

```

```

ShockwaveResistance = SHOCKWAVE_RESISTANCE_ALWAYS
KindOf = UNATTACKABLE PRELOAD CAN_CAST_REFLECTIONS INFANTRY
ARCHER_PATH_THROUGH_EACH_OTHER SCORE THROWN_OBJECT ATTACK_NEEDS_LINE_OF_SIGHT
NOTIFY_OF_PREATTACK

CommandPoints = 0

LocomotorSet
    Locomotor = FloatingLichtbringerLocomotor
    Condition = SET_NORMAL
    Speed = 0
End

Behavior = SubObjectsUpgrade ForgedBlade_Upgradel
    TriggeredBy = Upgrade_AllFactionUpgrade
    HideSubObjects = FireArrowTip
End

Behavior = SlavedUpdate ModuleTag_Slave
    DieOnMastersDeath = Yes
    MarkUnselectable = No
End

Behavior = AIUpdateInterface ModuleTag_03
    AutoAcquireEnemiesWhenIdle = Yes
End

WeaponSet
    Conditions = None
    Weapon = PRIMARY DwarvenBarracksArcherBow
End

VoiceCreated =
ElfWarriorVoiceSaluteBow_Slaved
VoiceFullyCreated =
ElfWarriorVoiceSaluteBow_Slaved

CommandSet = EmptyCommandSet

Behavior = WeaponSetUpgrade ModuleTag_FlammingAttack ;// Doesn't really use a
weaponset swap, the swap is in the weapon itself. But this helps debugging more
than a fake status bit upgrade
    TriggeredBy = Upgrade_GoodFortressFlamingMunitions
End
End

```

Mittlerweile kenne ich auch eine dritte Variante, um ein Slave-Object zu erstellen, mit speziellen Eigenschaften. Hierbei kann ein Creep-Object an eine bestimmte Einheit geheftet werden, ohne das dieses ständig in Bewegung ist. Ein Beispiel dafür wäre die Bestie des Bestienmeisters, die in Version 3.6.1 wie ein treuer Gefährte an der Seite ihres Meisters steht, diesem folgt und Feinde angreift, sobald diese in Reichweite sind. Damit wir dieses, von mir genannte, „Guard-System“ erlangen (eignet sich z.b. vorzüglich für „Wachposten“ an Gebäuden), müsst ihr folgenden Code in das zu bewachende Object einfügen:

```

Behavior = QueueProductionExitUpdate ModuleTag_11
    UnitCreatePoint = X:-25.0 Y:0.0 Z:0.0 ;(beispiel für
Erschaffungsort)
    NaturalRallyPoint = X:-25 Y:0.0 Z:0.0
    ExitDelay = 300

```

```

    NoExitPath          = Yes
End

Behavior = SpawnBehavior ModuleTag_SpawnFighters
    SpawnNumber         = 1
    InitialBurst        = 1
    SpawnTemplateName   = Objectname
    SpawnReplaceDelay   = 200000
    FadeInTime          = 1000
    SpawnInsideBuilding = Yes
End

```

Fügt nun in die ini des „Wächters“ folgenden Code ein:

```

Behavior = SlavedUpdate ModuleTag_Slave
    LeashRange          = 500 ;;;250
    GuardMaxRange       = 0
    GuardWanderRange    = 0
    AttackRange         = 200
    UseSlaverAsControlForEvaObjectSightedEvents = No // Not a creep
    GuardPositionOffset = X:0.0 Y:50 Z:0
    DieOnMastersDeath   = No
End

```

Dieser Abschnitt ist der entscheidende:
GuardPositionOffset

Der Punkt muss sich in einem mäßigen Abstand des zu bewachenden Objects befinden. Dort wird sich der Wächter immer abstellen, wenn er nicht mehr in Aktion ist (in meinem Fall die Bestie). Erst sobald Feinde in der Nähe sind, oder sich das zu bewachende Object bewegt, bewegt sich auch der „Wächter“.

Ich wünsche euch viel Spaß beim ausprobieren. Solltet ihr dieses Tutorial verwenden, bitte ich um einen Readme-Eintrag.

Ealendril der Dunkle