

Gotowe klasy - [UT'99] Authorizer - trigger na klucze by [Raven](#) dnia: 20 Kwiecien 2008, 13:02

Opis

Jest to zwykly trigger na klucze. W celu otwarcia go, gracz musi miec odpowiedni klucz. Klasa klucza musi byc podklasa **Inventory**.

Zmienne

Widoczne

class KeyClass: klasa klucza (najlepiej z inventory)

bool bDestroyKey: jesli wartosc zmiennej bedzie równa **true**, to klucz zostanie zniszczony zaraz po wejściu gracza na trigger

bool bCheckKeyOnceOnly: jesli wartosc zmiennej bedzie równa **true**, to trigger stanie sie normalny zaraz po znalezieniu klucza

bool bShowSuccessMessage: jesli wartosc zmiennej bedzie równa **true**, to trigger bedzie pokazywal informacje o znalezieniu klucza

bool bShowFailureMessage: jesli wartosc zmiennej bedzie równa **true**, to trigger bedzie pokazywal informacje o braku klucza

localized String SuccessMessage: wiadomosc o znalezieniu klucza

localized String FailureMessage: wiadomosc o braku klucza

Ukryte

bool bWasOpened: przyjmuje wartosc **true** tylko jesli klucz zostal znaleziony i zmienna

bCheckKeyOnceOnly ma takze wartosc **true**


```
//=====
// Authorizer: This Trigger can replace
// KeyMover (as base I use KeyMover available
// from http://chimeric.beyondunreal.com/tutorials/tut11.html).
//=====
// by Raven
// http://turniej.unreal.pl/rp
// for The Chosen One SP mod
//=====

class Authorizer extends Trigger;

var() class< Inventory > KeyClass;
var() bool bDestroyKey;
var() bool bCheckKeyOnceOnly;
var() bool bShowSuccessMessage;
var() bool bShowFailureMessage;
var() localized String SuccessMessage;
var() localized String FailureMessage;
var bool bWasOpened;

replication
{
    // Variables the server should send to the client.
    reliable if( Role==ROLE_Authority )
        bWasOpened;
}

function Touch( actor Other )
{
    local Inventory key;
    local actor A;

    if (Other.IsA('Pawn') && KeyClass != none && !bWasOpened)
    {
        key = Pawn(Other).FindInventoryType(KeyClass);

        if (key != none && Event != "" && !bCheckKeyOnceOnly)
            foreach AllActors( class 'Actor', A, Event )
                A.Trigger( Other, Other.Instigator );

        if(bDestroyKey)
            if(!bWasOpened) Pawn(Other).DeleteInventory(key);
        if(bCheckKeyOnceOnly)
            bWasOpened=true;
    }
    if(bWasOpened && Event != "")
        foreach AllActors( class 'Actor', A, Event )
            A.Trigger( Other, Other.Instigator );
}
```

```
    if(bShowSuccessMessage && PlayerPawn(Other) != none)
        PlayerPawn(Other).ClientMessage(SuccessMessage);
    if(bShowFailureMessage && !bWasOpened)
        PlayerPawn(Other).ClientMessage(FailureMessage);
}
```

defaultproperties

```
{
    bShowFailureMessage=True
    SuccessMessage="Access granded."
    FailureMessage="You need a key to open this door."
}
```