Inhalt

Verwendung

Manchmal ist es notwendig, den Spieler etwas einzuschränken.

Du kannst dem Spieler den Bau von Gebäuden verbieten.
Des weiteren kannst auch den Abriss einschränken.

Funktionen

Bau verbieten

```
API.RestrictBuildingCustomFunction
API.RestrictBuildingTypeInTerritory
API.RestrictBuildingTypeInArea
API.RestrictBuildingCategoryInTerritory
API.RestrictBuildingCategoryInArea
API.RestrictRoadCustomFunction
API.RestrictTrailInTerritory
API.RestrictTrailInArea
API.RestrictStreetInTerritory
API.RestrictStreetInArea
API.DeleteRestriction
```

Abriss verbieten

```
API.ProtectBuildingCustomFunction
API.ProtectBuildingTypeInTerritory
API.ProtectBuildingTypeInArea
API.ProtectBuildingCategoryInTerritory
API.ProtectBuildingCategoryInArea
API.ProtectNamedBuilding
API.DeleteProtection
```

Priorität

Die Einschränkungen für den Spieler haben unterschiedliche Priorität.
Manche Einstellungen werden vor anderen geprüft.
Die Hierarchie kann sich nach zwei Hauptkriterien richten.

Nach Art des Geltungsbereich:

1. Custom-Funktion
2. Bereich
3. Territorium

Nach Art des Gebäudes:

4. Custom-Funktion
5. Skriptname des Gebäudes
6. Typ des Gebäudes
7. Kategorie des Gebäudes

Benutzerdefinierte Bedingungen

Die vorgegebenen Funktionen reichen oft nicht aus.

Du kannst eigene Bedingungen definieren, indem du Custom-Funktionen benutzt.

Die Funktionen müssen im lokalen Skript ausgeführt werden!

Gebäude beschränken

Den Bau von Gebäudetypen beschränken.

```
local MyCustomRestriction = function(_PlayerID, _Type, _X, _Y)
    -- Bedingungen für das Verbot prüfen
    if AnythingIWant then
        return true;
    end
    -- Alles andere bleibt unbehelligt
    return false;
end
MyRestrictionID = API.RestrictBuildingCustomFunction(1, MyCustomRestriction);
```

Straßen beschränken

Den Bau von Straßen beschränken.

```
local MyCustomRestriction = function(_PlayerID, _IsTrail, _X, _Y)
    -- Bedingungen für das Verbot prüfen
    if AnythingIWant then
        return true;
    end
    -- Alles andere bleibt unbehelligt
    return false;
end
MyRestrictionID = API.RestrictRoadCustomFunction(1, MyCustomRestriction);
```

Abriss beschränken

Den Abriss von Gebäuden beschränken.

```
local MyCustomProtection = function(_PlayerID, _BuildingID, _X, _Y)
    -- Bedingungen für das Verbot prüfen
    if AnythingIWant then
        return true;
    end
    -- Alles andere bleibt unbehelligt
    return false;
end
MyProtectionID = API.ProtectBuildingCustomFunction(1, MyCustomProtection);
```

Beispiel: Militärgebäude beschränken

Wenn Du die Nachschubwege künstlich verlängern willst, kannst Du die Kasernen beschränken.
Hier können Militärgebäude nur auf einem Territorium gebaut werden.

```
-- Einschränkung erstellen
local RestrictMilitaryBuildings = function(_PlayerID, _Type, _X, _Y)
    -- Die Bedingung wird nur für Militärgebäude geprüft.
    if _Type == Entities.B_Barracks
    or _Type == Entities.B_BarracksArchers
    or _Type == Entities.B_SiegeEngineWorkshop
    or _Type == Entities.B_SwordSmith
    or _Type == Entities.B_BowMaker then
        -- Militärgebäude können nur auf Territorium 8 gebaut werden.
        if Logic.GetTerritoryAtPosition(_X, _Y) ~= 8 then
            return true;
        end
    end
    -- Alles andere kann gebaut werden.
    return false;
end
-- Einschränkung für Spieler 1 aktivieren
RestrictionID = API.RestrictBuildingCustomFunction(1, RestrictMilitaryBuildings);
```

Beispiel: Die Wirtschaft kastrieren

Vielleicht möchtest Du, dass der Spieler beim Bauen der Stadt mehr nachdenken muss.
Hier können nur 3 Rohstoffgebäude pro Territorium gebaut werden.
Außerdem können Stadtgebäude nur auf Territorium 12 gebaut werden.
Als Bonus oben drauf müssen Bienenstöcke bei Imkern stehen.

```
-- Einschränkung erstellen
local RestrictEconomy = function (_PlayerID, _Type, _x, _y)
    -- Rohstoffgebäude sind auf 3 pro Gebiet beschränkt.
    if Logic.IsEntityTypeInCategory(_Type, EntityCategories.OuterRimBuilding) == 1 then
        local TerritoryID = Logic.GetTerritoryAtPosition(_x, _y);
        local Buildings = API.SearchEntitiesOfCategoryInTerritory(
            TerritoryID, EntityCategories.OuterRimBuilding, _PlayerID
        );
        return #Buildings >= 3;
    end
    -- Stadtgebäude können nur auf Territorium 12 gebaut werden.
    if Logic.IsEntityTypeInCategory(_Type, EntityCategories.CityBuilding) == 1 then
        return Logic.GetTerritoryAtPosition(_x, _y) == 12;
    end
    -- Bienenstöcke müssen in der Nähe von Imkern stehen
    if _Type == Entities.B_Beehive then
        local n, OPID = Logic.GetPlayerEntitiesInArea(
            _PlayerID, Entities.B_Beekeeper, _x, _y, 2500, 1
        );
        if n == 0 then
            return true;
        end
    end
    -- Alles andere kannn gebaut werden
    return false;
end
-- Einschränkung für alle Spieler aktivieren
API.RestrictBuildingCustomFunction(-1, RestrictEconomy);
```

Beispiel: Palisaden nur um Außenposten

Mauern und Palisaden sind viel zu stark und sollten beschränkt werden.
Hier können Palisaden nur um Außenposten herum gebaut werden.

```
-- Einschränkung erstellen
local RestrictPalisadeToOutpost = function (_PlayerID, _Type, _x, _y)
    -- Die Bedingung wird nur für Palisaden geprüft.
    if Logic.IsEntityTypeInCategory(_Type, EntityCategories.PalisadeSegment) == 1 then
        -- Es muss ein Außenposten in der Nähe sein.
        local n, OPID = Logic.GetPlayerEntitiesInArea(
            _PlayerID, Entities.B_Outpost_ME, _x, _y, 1500, 1
        );
        if n == 0 then
            return true;
        end
    end
    -- Alles andere kannn gebaut werden
    return false;
end
-- Einschränkung für alle Spieler aktivieren
API.RestrictBuildingCustomFunction(-1, RestrictPalisadeToOutpost);
```