

Inhalt

Verwendung.....	2
Grundlagen.....	2
Seiten erstellen.....	3
Normale Seite.....	3
Vereinfachte Seite.....	3
Entscheidungen.....	4
Verzweigungen.....	4
Seitennamen.....	4
Dialogoptionen.....	4
Beispiele.....	5
Beispiel: Audienz beim Kastellan.....	5
Beispiel: Handelsware auswählen.....	6

Verwendung

Komplexe Dialoge solltest Du nicht mit Quests umsetzen.

Das Dialog System bietet eine mehr als gute Alternative dafür. Vor allem wenn Dir Briefings zu „unpersönlich“ sind. Dialoge verwenden die Portraits, genau wie bei Quests. Dies ohne die damit verbundenen Nachteile. Das Portrait kann auch ausgeblendet werden.

Dialoge kannst Du an Quests anbinden – musst Du aber nicht.

Grundlagen

Dialoge stehen innerhalb einer Lua-Funktion im globalen Skript.

Die folgende Funktion kannst Du als Kopiervorlage verwenden.

```
function Dialog1(_Name, _PlayerID)
    local Dialog = {
        DisableFow = true,
        DisableBoderPins = true,
    };
    local AP, ASP = API.AddDialogPages(Dialog);

    -- Aufrufe von AP oder ASP um Seiten zu erstellen

    Dialog.Starting = function(_Dialog)
        -- Mach was tolles hier, wenn es anfängt.
    end
    Dialog.Finished = function(_Dialog)
        -- Mach was tolles hier, wenn es endet.
    end
    API.StartDialog(Dialog, _Name, _PlayerID);
end
```

Die Funktion muss die Parameter `_Name` und `_PlayerID` entgegen nehmen.

Zuerst erstellst du eine Tabelle für den Dialog. Nenne sie z.B. `Dialog`. Hier kannst Du einige Einstellungen für den Dialog vornehmen. Siehe dazu die Dokumentation.

Die Tabelle wird zusammen mit den Parametern an das Dialog System übergeben.

Mit der Funktion `API.AddDialogPages` werden die Funktionen für die Seiten erzeugt.

Mit `AP` und `ASP` erstellst Du die einzelnen Seiten.

Die Seiten zeigen den gesprochenen Text an. Sie bestimmen zudem die Position der Kamera und das verwendete Portrait. Du kannst den Spieler ebenfalls Entscheidungen treffen lassen.

Zu Beginn und Ende des Dialog kannst du Funktionen ausführen lassen.

- `Dialog.Starting` wird zu Beginn, vor der ersten Seite, ausgeführt.
- `Dialog.Finished` wird am Ende, nach der letzten Seite, ausgeführt.

Abschließend startest du den Dialog mit `API.StartDialog`.

Seiten erstellen

Normale Seite

Eine normale Seite wird mit der Funktion AP erstellt.

```
AP {  
    -- Name im Dialogfenster  
    Title      = "Marcus",  
    -- Text im Dialogfenster  
    Text       = "Der Himmel ist so grau, weil alles Blaue schon herunter "..  
                "gelogen wurde.",  
    -- Verwende Portrait von Spieler 1  
    Actor      = 1,  
    -- Position der Kamera  
    -- (Entfernung und Rotation können ebenfalls angepasst werden)  
    Position   = "marcus",  
    -- Kameranahsicht verwenden  
    DialogCamera = true,  
};
```

In der Dokumentation kannst du alle Attribute nachlesen.

Vereinfachte Seite

Manchmal ist die normale Seite zu sperrig. Wenn Du z.B. nur ein paar Seiten normalen Dialog erzeugen willst. So kannst du einfach viel Platz sparen.

Die vereinfachte Seite erstellst Du mit ASP erstellen.

```
-- Sender, Position, Titel, Text, Dialogkamera  
ASP(1, "marcus", "Marcus", "Der Himmel ist so grau, weil alles Blaue "..  
    "schon herunter gelogen wurde.", true);  
ASP(2, "alandra", "Alandra", "Könnt Ihr bitte aufhören, diesen alten "..  
    "Pfaffen zu rezitieren.", true);
```

In der Dokumentation kannst du alle Parameter nachlesen.

Entscheidungen

Verzweigungen

Du kannst den Spieler in Dialogen Entscheidungen fällen lassen.

Eine Entscheidung eröffnet eine Verzweigung. Der Spieler wählt eine von mehreren Optionen. Sie entscheidet, welchem Pfad der Dialog folgt.

Die Anzahl an Entscheidungen und Verzweigungen ist unbegrenzt.

Verzweigungen werden mit leeren Seiten oder Weiterleitungen abgegrenzt.

```
AP();
```

Dies ist eine leere Seite.

```
AP("AnyPageName");
```

Dies ist eine Weiterleitung auf eine benannte Seite.

Zur Benennung von Seiten kommen wir gleich.

Seitennamen

Damit Verzweigungen funktionieren, benötigen die Seiten Namen.

Ein Name identifiziert eine Seite. Du kannst keine zwei Seiten gleich benennen.

```
AP {  
    Name = "NameOfPage",  
    -- Weitere Optionen  
};
```

So benennst Du eine Seite.

```
ASP("NameOfPage", ...)
```

So benennst Du eine vereinfachte Seite.

Dialogoptionen

Du kannst nur bei AP Optionen für Entscheidungen angeben.

Es gibt keine Begrenzung für die Anzahl von Optionen.

Der Text sollte jedoch möglichst kurz sein.

Die Optionen werden mit dem Attribut MC angelegt.

```
AP {  
    Title      = "Hakim",  
    Text       = "Der Feind ist klüger als es zuerst schien. Wie sollen "...  
                "wir vorgehen, Milord?",  
    Actor      = 1,  
    Position   = "hakim",  
    DialogCamera = true,  
    MC         = {  
        -- Anzeigetext, Seite zu der gesprungen wird  
        {"Mit List und Tücke.", "TheSlyWay"},  
        {"Ehrenhaft wie echte Männer.", "TheHonoredWay"},  
        {"ROHE GEWALT!!", "TheDumbWay"},  
    }  
};
```

Je nach Wahl des Spielers werden verschiedene Pfade beschritten.

1. „Mit List und Tücke.“ springt zur Seite „TheSlyWay“
2. „Ehrenhaft wie echte Männer.“ springt zur Seite „TheHonoredWay“
3. „ROHE GEWALT!!“ springt zur Seite „TheDumbWay“

Beispiele

Beispiel: Audienz beim Kastellan

Wir erstellen einen normalen Dialog ohne Verzweigungen.

Angenommen, du willst den Helden eine Audienz bei einem Stadthalter haben lassen. Dies könnte bpsw. wie folgt aussehen:

```
function AudienzBeiKastellan(_Name, _PlayerID)
    local Dialog = {
        DisableFow = true,
        DisableBoderPins = true,
    };
    local AP, ASP = API.AddDialogPages(Dialog);

    ASP(3, "hero3", "Stadthalter", "Schaut an, ein fremder Edelmann besucht "..
        "mein beschauliches Städtchen. Was mag er von mir wollen?", true);
    ASP(1, "marcus", "Marcus", "Nun ich fragte mich, ob seine Lordschaft es "..
        "in Betracht zieht, mit mir Handel zu treiben.", true);
    ASP(3, "hero3", "Stadthalter", "Wo kommen wir denn hin, wenn ich jedem "..
        "dahergelaufenen Edelmann meine Waren verkaufe? Was habt Ihr denn "..
        "für mich getan?", true);
    ASP(1, "marcus", "Marcus", "Was verlangt Ihr für Eure Gunst?", true);
    ASP(3, "hero3", "Stadthalter", "Nun, da gäbe es die eine oder andere "..
        "Kleinigkeit, die Ihr für mich erledigen könntet...", true);

    API.StartDialog(Dialog, _Name, _PlayerID);
end
```

Später hängst Du den Dialog z.B. über einen Reward an einen Auftrag.

```
Reward_Dialog("D1P3", "AudienzBeiKastellan")
```

Sobald der Auftrag erfüllt wird, startet der Dialog.

Beispiel: Handelsware auswählen

Nun erstellen wir einen Dialog mit Verzweigungen.

Der Spieler kann sich entscheiden, welche Handelswaren angeboten werden sollen.

```
function SelectDesiredTradeGoods(_Name, _PlayerID)
    local Dialog = {
        DisableFow = true,
        DisableBoderPins = true,
    };
    local AP, ASP = API.AddDialogPages(Dialog);

    ASP(3, "trader", "Händler", "Lasst mich Euch ein Angebot machen, dass "..
        "Ihr nicht ablenhen könnt..", true);
    ASP(3, "trader", "Händler", "Erlaubt mir, das Angebot nur für Euch zu "..
        "personalisieren. Welche Waren interessieren Euch mehr?", true);

    -- Der Spieler muss auf dieser Seite eine Option wählen, um fortzufahren.
    AP {
        -- Die Seite bekommt einen Namen, um später die Antwort zu erhalten.
        Name          = "Selection",

        Title          = "Kestral",
        Text           = "Hm... Was soll ich wählen?",
        Actor          = 1,
        Position       = "kestral",
        DialogCamera   = true,
        MC             = {
            {"Ökonomischer Fokus", "Economy"},
            {"Militärischer Fokus", "Military"},
        }
    };

    -- Springt hier her, wenn die "Economy"-Option gewählt wird
    ASP("Economy", 1, "kestral", "Kestral", "Wenn ich eine starke Wirtschaft "..
        " aufbaue, kommt der Rest wie von ganz allein. Ich wünsche Angebote, "..
        " die meine Wirtschaft stärken.", true);
    -- Springe weiter zur letzten Seite
    AP("Final");

    -- Springt hier her, wenn die "Military"-Option gewählt wird
    ASP("Military", 1, "kestral", "Kestral", "Wenn ich alle meine Feinde "..
        " vernichte, kann ich mich danach um wie Wirtschaft sorgen. Macht "..
        " mir Angebote für ein starkes Heer.", true);

    -- Hier kann der Dialog einfach weiter laufen
    ASP("Final", 3, "trader", "Händler", "Sehr wohl! Ich werde Euch Waren "..
        "vorschlagen, die Euren Bedürfnissen entsprechen.", true);

    Dialog.Finished = function(_Dialog)
        -- Funktion entsprechend der Antwort aufrufen.
        local Selected = _Dialog:GetPage("Selection"):GetSelected();
        if Selected == 1 then
            SetupEconomyOffers();
        elseif Selected == 2 then
            SetupMilitaryOffers();
        end
    end
    API.StartDialog(Dialog, _Name, _PlayerID);
end
```

Später hängst Du den Dialog z.B. über einen Reward an einen Auftrag.

```
Reward_Dialog("SelectGoods", "SelectDesiredTradeGoods")
```

Sobald der Auftrag erfüllt wird, startet der Dialog.