

## Inhalt

Verwendung.....	2
Grundlagen.....	2
Seiten erstellen.....	3
Normale Seite.....	3
Vereinfachte Seite.....	3
Entscheidungen.....	4
Grundlagen.....	4
Seitennamen.....	4
Verzweigungen.....	4
Kameraanimationen.....	5
Grundlagen.....	5
Beispiele.....	5
Zusatzoptionen.....	5
Parallaxe.....	6
Grafiken.....	6
Angabe.....	6
Beispiele.....	6
Zusatzoptionen.....	7
Beispiele.....	8
Beispiel: Audienz beim Kastellan.....	8
Beispiel: Heldenwahl.....	9

## Verwendung

Komplexe Dialoge solltest Du nicht mit Quests umsetzen.

Das Briefing System bietet eine mehr als gute Alternative dafür. Du kannst auch Kamera-und Parallaxanimationen einsetzen, damit Dein Briefing noch besser wird.

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

## Grundlagen

Briefings stehen innerhalb einer Lua-Funktion im globalen Skript.

Die folgende Funktion kannst Du als Kopiervorlage verwenden.

```
function Briefing1(_Name, _PlayerID)
    local Briefing = {
        -- Hier können verschiedene Konfigurationen vorgenommen werden.
    };
    local AP, ASP = API.AddBriefingPages(Briefing);

    -- Aufrufe von AP oder ASP um Seiten zu erstellen

    Briefing.Starting = function(_Data)
        -- Mach was tolles hier, wenn es anfängt.
    end
    Briefing.Finished = function(_Data)
        -- Mach was tolles hier, wenn es endet.
    end
    -- Das Briefing wird gestartet
    API.StartBriefing(Briefing, _Name, _PlayerID);
end
```

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

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

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

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

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

Die Seiten zeigen den gesprochenen Text an. Sie können zudem die Kameraeinstellungen bestimmen. Du kannst den Spieler ebenfalls Entscheidungen treffen lassen.

Zu Beginn und Ende des Briefings 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.StartBriefing`.

## Seiten erstellen

### Normale Seite

Eine normale Seite wird mit der Funktion AP erstellt.

```
AP {  
    -- Text im Titelfeld  
    Title      = "Marcus",  
    -- Text im Textfeld  
    Text       = "Der Himmel ist so grau, weil alles Blaue schon herunter "..  
                "gelogen wurde.",  
    -- Position der Kamera  
    Position   = "marcus",  
    -- Rotation der Kamera  
    Rotation   = 30,  
    -- Kameransicht 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.

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

In der Dokumentation kannst du alle Parameter nachlesen.

## Entscheidungen

### Grundlagen

Du kannst den Spieler in Briefings Entscheidungen fällen lassen.

Eine Entscheidung eröffnet eine Verzweigung. Der Spieler wählt eine von mehreren Optionen. Sie entscheidet, welchem Pfad das Briefing 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.

### Verzweigungen

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?",  
    Position    = "hakim",  
    Rotation    = 30,  
    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“

## Kameraanimationen

### Grundlagen

Du kannst die Animation der Kamera von den Seiten entkoppeln.

Entkoppelte Animationen haben einen entscheidenden Vorteil. Sie ermöglichen nahtlose Übergänge zwischen mehreren Seiten.

Eine Seite darf keine Positionsangabe haben.

Animationen werden in der Briefing Tabelle angelegt.

```
Briefing.PageAnimations = {}
```

Animationen müssen immer noch eine Seite als Startpunkt haben. Du kannst mehrere Animationen pro Seite hinzufügen. Sie laufen nacheinander ab, bis alle abgespielt wurden.

Die Startseite der Animation muss einen Namen haben.

Es gibt zwei Möglichkeiten eine Position zu definieren.

1. Relativ. Du gibst Position, Rotation, Entfernung und Winkel an.
2. Vektor. Du gibst die XYZ-Koordinate von Position und Blickziel an.

Es muss je eine Animationsdauer und Startposition angegeben werden. Die Endposition kannst Du weglassen. Dann bleibt die Kamera am Start stehen.

### Beispiele

```
Briefing.PageAnimations = {
  ["Page1"] = {
    -- Relativdarstellung
    -- Dauer, PositionStart, RotationStart, ZoomStart, AngleStart
    -- PositionEnde, RotationEnde, ZoomEnde, AngleEnde
    {30, "pos4", -60, 2000, 35,
     "pos4", -30, 2000, 25},
    -- ... weitere Animationen
  },
}
```

Ein Beispiel für die relative Koordinatenangabe.

```
Briefing.PageAnimations = {
  ["Page1"] = {
    -- Vektordarstellung
    -- Dauer, {Position1, Höhe}, {LookAt1, Höhe},
    -- {Position2, Höhe}, {LookAt2, Höhe}
    {30, {"pos2", 500}, {"pos4", 0},
     {"pos7", 1000}, {"pos8", 0}},
    -- Hier können weitere Animationen folgen...
  },
}
```

Ein Beispiel für die Angabe von Vektoren.

### Zusatzoptionen

Laufende Animationen können durch **Clear** gelöscht werden.

```
Briefing.PageAnimations = {
  ["Page1"] = {
    Clear = true,
    ...
  }
}
```

Mit **Repeat** werden die Animationen in Endlosschleife gesetzt.

```
Briefing.PageAnimations = {
  ["Page1"] = {
    Repeat = true,
    ...
  }
}
```

## Parallaxe

### Grafiken

Du kannst bis zu 4 Hintergründe übereinander anzeigen lassen.

Diese Hintergründe werden hinter allen Elementen des Thronerom angezeigt.

Die Grafiken müssen in die Map gepackt werden.

```
mapname.s6xmap.unpacked
|-- maps/externalmap/mapname
|-- graphics/*
```

Am Besten verwendest du einen Grafiken-Ordner in der Map.

Eine einmal geladene Grafik verbleibt im Speicher. Sie geändert noch einmal zu laden, funktioniert nicht. Um den Cache zu leeren, musst Du das Spiel beenden.

### Angabe

Parallaxe werden in der Briefing Tabelle angelegt.

```
Briefing.PageParallax = {}
```

Ein Parallax hat ebenfalls eine Seite als Startpunkt.

Es muss je eine Animationsdauer und Startposition angegeben werden. Die Endposition kannst Du weglassen. Dann bleibt der Bildausschnitt stehen.

Du kannst den angezeigten Bereich über die Angabe von Koordinaten verändern.

Diese werden als UV-Koordinaten bezeichnet.

Das UV-Mapping ist das Anpassen einer Textur auf ein Objekt. Hier wird eine Textur auf eine zweidimensionale Ebene projiziert. Du kannst entweder die vollständige Grafik oder einen Ausschnitt der Grafik anzeigen.

Du kannst die UV-Koordinaten relativ oder absolut angeben.

Für letzteres musst du die Größe der Grafik wissen.

Das erste Paar gibt die obere linke Ecke an.

Das zweite Paar gibt die untere rechte Ecke an.

### Beispiele

```
Briefing.PageParallax = {
  ["Page1"] = {
    -- Bilddatei, Anzeigedauer,
    -- U0Start, V0Start, U1Start, V1Start, AlphaStart,
    -- U0End, V0End, U1End, V1End, AlphaEnd
    {"C:/IMG/Bild1.png", 60,
     0, 0, 0.8, 1, 255,
     0.2, 0, 1, 1, 255},
    -- Hier können weitere Einträge folgen...
  },
}
```

Ein Beispiel für die relativen UV-Koordinaten.

Die Größe der Grafik ist hier egal. Es werden Werte zwischen 0 und 1 als Koordinaten verwendet. Sie stehen für 0 bis 100% der Grafik.

```
Briefing.PageParallax = {  
  ["Page1"] = {  
    -- Bilddatei, Anzeigedauer,  
    -- U0Start, V0Start, U1Start, V1Start, AlphaStart,  
    -- U0End, V0End, U1End, V1End, AlphaEnd  
    {"C:/IMG/Bild2.png", 60,  
     0, 1366, 768, 255,  
     0, 1622, 768, 255},  
    -- Hier können weitere Einträge folgen...  
  },  
}
```

Ein Beispiel für die absoluten UV-Koordinaten.

Die Grafik in diesem Beispiel ist 1622x768 Pixel groß. Der Bildausschnitt stellt einen Bereich von 1366x768 Pixel dar. Dabei wandert der Ausschnitt von links nach rechts.

## Zusatzoptionen

Parallaxe können durch `Clear` gelöscht werden.

```
Briefing.PageParallax = {  
  ["Page1"] = {  
    Clear = true,  
    ...  
  },  
}
```

## Beispiele

### Beispiel: Audienz beim Kastellan

Wir erstellen ein normales Briefing ohne Verzweigungen.

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

```
function Briefing2_CastellanMeeting(_Name, _PlayerID)
    local Briefing = {
        DisableFow = true,
        DisableBoderPins = true,
    };
    local AP, ASP = API.AddBriefingPages(Briefing);

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

    API.StartBriefing(Briefing, _Name, _PlayerID);
end
```

Später hängst Du das Briefing z.B. über einen Reward an einen Auftrag.

```
Reward_Briefing("Briefing2", "Briefing2_CastellanMeeting")
```

Sobald der Auftrag erfüllt wird, startet das Briefing.



## Beispiel: Heldenwahl

Nun erstellen wir einen Dialog mit Verzweigungen.

Der Spieler kann hier entscheiden, mit welchem Helden er spielen will.

```
function Briefing01_ChooseKnight(_Name, _PlayerID)
    local Briefing = {};
    local AP, ASP = API.AddBriefingPages(Briefing);

    AP {
        Name          = "Selection",
        Title          = "Wählt Euren Ritter!",
        Position       = "Campfire",
        DialogCamera   = true,
        MC = {
            {"Marcus", Briefing01_ChooseKnight_SelectHero},
            {"Alandra", Briefing01_ChooseKnight_SelectHero},
            {"Hakim", Briefing01_ChooseKnight_SelectHero},
        },
    };

    AP {
        Name          = "Confirm",
        Title          = "Seid Ihr sicher?",
        Position       = "Campfire",
        DialogCamera   = true,
        MC = {
            {"Marcus", Briefing01_ChooseKnight_ConfirmSelection},
            {"Alandra", Briefing01_ChooseKnight_ConfirmSelection},
            {"Hakim", Briefing01_ChooseKnight_ConfirmSelection},
        },
    };

    -- Selbstbeweihräucherung der Helden
    -- Anschließend wird zur Seite "Confirm" gesprungen
    ASP("MarcusPage", "Marcus", "Wählt mich, und ich werde die Feinde des "..
        "Reiches in Eurem Namen vernichten!", false, "marcus");
    AP("Confirm");
    ASP("AlandraPage", "Alandra", "Etwas weibliche Vorsicht ist in dieser "..
        "Situation männlichem 'Hauruck' überlegen.", false, "alandra");
    AP("Confirm");
    ASP("HakimPage", "Hakim", "Sollte Eure Wahl auf mich fallen, werde ich "..
        "den Feind mit List anstatt Gewalt bekämpfen.", false, "hakim");
    AP("Confirm");

    -- Danksagungen der Helden
    -- Das Briefing endet nach jeder Danksagung eines Helden
    ASP("MarcusThanks", "Marcus", "Vortrefflich! Ich werden sofort beginnen, "..
        "Euch Ehre zu erweisen!", false, "marcus");
    AP();
    ASP("AlandraThanks", "Alandra", "Eure Weise Entscheidung wird unserer "..
        "Sache zum Vorteil gereichen.", false, "alandra");
    AP();
    ASP("HakimThanks", "Hakim", "Ich habe schon einen Plan. Habt Dank, dass "..
        "ich ihn Euch vorführen darf.", false, "hakim");

    Briefing.Finished = function(_Data)
        StartGame();
    end
end
```

Später hängst Du das Briefing z.B. über einen Reward an einen Auftrag.

```
Reward_Briefing("Briefing01", "Briefing01_ChooseKnight")
```

Sobald der Auftrag erfüllt wird, startet das Briefing.

Diese Funktion handhabt die Auswahl.

Es werden Funktionen aufgerufen, die den Helden austauschen. (Nicht in diesem Beispiel enthalten)  
Danach wird zur entsprechenden Textseite gesprungen.

```
function Briefing01_ChooseKnight_SelectHero(_PlayerID, _PageID, _OptionID)
    gvMission.SelectedHero = _OptionID;
    if _OptionID == 2 then
        PrepareStartWithAlandra();
        return "AlandraPage";
    elseif _OptionID == 3 then
        PrepareStartWithHakim();
        return "HakimPage";
    end
    PrepareStartWithMarcus();
    return "MarcusPage";
end
```

Hier wird entweder die Auswahl bestätigt oder zurück zur Auswahl gesprungen.

Wird die Auswahl bestätigt, beginnt das Spiel.

```
function Briefing01_ChooseKnight_ConfirmSelection(_PlayerID, _PageID, _OptionID)
    if _OptionID == 1 then
        if gvMission.SelectedHero == 2 then
            return "AlandraThanks";
        elseif gvMission.SelectedHero == 3 then
            return "HakimThanks";
        end
        return "MarcusThanks";
    end
    return "Selection";
end
```