

# マチキャラ コンテンツ開発ガイド

## 第 4.0 版

平成 22 年 11 月 8 日  
株式会社 NTT ドコモ

---

## 目次

はじめに .....	1
概要 .....	1
必要となる知識 .....	1
マチキャラコンテンツ開発者ガイダンス .....	2
1. マチキャラとは .....	2
2. マチキャラの種類 .....	2
2.1. マチキャラデータ .....	2
2.2. マチキャラを構成するファイル .....	2
3. マチキャラのバージョン .....	3
4. マチキャラでできる演出 .....	4
4.1. 各種イベントによるアクションの実行 .....	4
4.2. マスコットの差し替え .....	6
4.3. マスコットの着せ替え .....	6
4.4. バルーン表示 .....	6
4.5. ポップアップ表示 .....	6
4.6. Z バッファ・パースペクティブコレクションでの描画 .....	6
4.7. 3D に特化した演出 .....	6
4.8. ネットワークアップデート .....	6
4.8.1. CFDID .....	6
4.8.2. ネットワークアップデート動作概要 .....	7
4.8.3. 更新契機 .....	7
4.8.4. ネットワークアップデート動作概要 .....	7
5. マチキャラデータの構成 .....	10
6. マチキャラコンテンツ作成の流れ .....	11
6.1. 2D マチキャラの場合 .....	11
6.2. 3D マチキャラの場合 .....	13
6.3. コンテンツの配信 .....	15
6.3.1. マチキャラ Ver.1.0、2.0 の場合 .....	15
6.3.2. マチキャラ Ver.3.0 以降の場合 .....	15
6.3.3. RangeRequest 機能の HTTP リクエスト/レスポンス .....	16
6.4. マチキャラコンテンツの DRM .....	18
7. マチキャラコンテンツ作成における制限事項 .....	19
8. ビヘイビア・モータの概念 .....	27
8.1. キャラクタの動作 .....	27
8.2. アクションの合成 .....	30
8.3. パーツの状態 .....	32

8.4. モータとは.....	34
8.5. ビヘイビアとは .....	36
8.6. ビヘイビアとモータの詳細.....	37
9. マチキャラコンテンツ作成におけるガイドライン.....	54
9.1. アニメーションの定義 .....	54
9.1.1. ボーンアニメーション(3D) .....	54
9.1.2. テクスチャアニメーション(3D) .....	54
9.1.3. GIF アニメーション .....	54
9.1.4. バルーンアニメーション .....	54
9.2. 3D マチキャラのキャラクタ作成.....	55
9.2.1. モータ(パーツ)の考え方 .....	55
9.2.2. アクション.....	56
9.2.3. パーツ分け .....	56
9.2.4. アニメーションの競合 .....	56
9.3. 2D マチキャラのキャラクタ作成.....	58
9.3.1. 2D アニメーションの制御.....	58
9.4. キャラクタの動作設定.....	59
9.4.1. イベント設定 .....	59
9.4.2. イベント状態の解除.....	59
9.4.3. イベントの実行制限.....	59
9.5. キャラクタの移動.....	61
9.5.1. 自動移動 .....	61
9.5.2. 定型移動 .....	61
9.6. バルーン表示.....	64
9.6.1. 文字列置換機能 .....	66
9.7. ポップアップ表示 .....	67
9.8. 着せ替え表現 .....	68
9.8.1. 基本 .....	68
9.8.2. 時間反映表現.....	68
9.8.3. 隠しマスコット.....	68
9.9. マスコット差替え.....	70
9.10. まとめ .....	71
スクリプトリファレンス .....	73
S.1. 構文表記法 .....	74
S.1.1. 構文のルール.....	74
S.1.2. 見出語 .....	74

---

S.1.3. 選択肢 .....	74
S.1.4 繰り返し.....	74
S.2. 字句要素.....	75
S.2.1. 文字集合 .....	75
S.2.1.1. アルファベット .....	75
S.2.1.2. 数字 .....	75
S.2.1.3. 図形文字.....	75
S.2.1.4. 特殊文字.....	75
S.2.1.5. 水平タブ .....	76
S.2.1.6. 改行文字.....	76
S.2.1.7. コメント .....	76
S.2.1.8. トークン .....	76
S.2.1.9. トークンの回避 .....	76
S.2.1.10. その他.....	76
S.3. 構文規則.....	77
S.3.1. ビヘイビアファイル .....	78
S.3.1.1. model セクション .....	80
S.3.1.2. motor セクション .....	81
S.3.1.3. texture セクション .....	82
S.3.1.4. map セクション .....	83
S.3.1.5. move セクション .....	84
S.3.1.6. balloon セクション .....	85
S.3.1.7. dress セクション .....	86
S.3.1.8. event セクション .....	88
S.3.1.9. コマンド詳細( event ) .....	92
S.3.1.10. ビヘイビアファイル設定例 .....	96
S.3.2. モータファイル .....	101
S.3.2.1. イベント(motor) .....	103
S.3.2.2. コマンド詳細(motor) .....	104
S.3.2.3. モータファイル設定例 .....	105
S.3.3. マチキャラメイクファイル.....	108
S.3.3.1. version セクション .....	110
S.3.3.2. distr セクション .....	111
S.3.3.3. id セクション .....	112
S.3.3.4. title セクション .....	113
S.3.3.5. url セクション .....	114

---

S.3.3.6. popup セクション .....	115
S.3.3.7. contentid セクション .....	116
S.3.3.8. fileid セクション .....	117
S.3.3.9. mmdurladdress セクション .....	118
S.3.3.10. mmdurlquery セクション .....	119
S.3.3.11. mascot セクション .....	120
S.3.3.12. cfdurl セクション .....	123
S.3.3.13. cfdtime セクション .....	124
S.3.3.14. filename セクション .....	125
S.3.3.15. マチキャラメイクファイル設定例 .....	126
<b>S.3.4. バルーンデータファイル .....</b>	<b>129</b>
S.3.4.1. pallet セクション .....	130
S.3.4.2. image セクション .....	131
S.3.4.3. message セクション .....	133
S.3.4.4. バルーンデータファイル設定例 .....	135
<b>S.3.5. 定型移動データファイル .....</b>	<b>137</b>
S.3.5.1. 定型移動データファイル設定例 .....	137
<b>S.3.6. ポップアップデータファイル .....</b>	<b>138</b>
S.3.6.1. ポップアップデータファイル設定例 .....	140

## 商標について

「iモード」「マチキャラ」「iコンシェル」「トルカ」は株式会社NTTドコモの商標または登録商標です。

その他記載された会社名、製品名などは該当する各社の商標または登録商標です。

## 改版履歴

版数	変更箇所	変更内容
1.0		新規作成
1.1	3.3	誤記修正
2.0	全体	マチキャラ Ver.2.0 の情報を追加
	全体	見直しによりドキュメント構成を変更
	全体	説明をわかりやすく変更。
	全体	誤記など修正
	2	マチキャラピクチャの説明を削除
	3	マチキャラのバージョンを追加
	4	マチキャラでできる演出を追加
	5	マチキャラデータの構成を追加、以降章番号繰り下げ
	6.1	⑤ マチキャラコンテンツで設定する色が実際の携帯端末で表現されない場合の注意書きを追加
	6.2	⑦ マチキャラコンテンツで設定する色が実際の携帯端末で表現されない場合の注意書きを追加
	7	使用可能な 3D ツールの ExporterPlugin の説明を修正
	7	作成する 3D モデルの推奨ポリゴン数を 750 から 1000 に変更
	7	モデルのポリゴン属性ににブレンディングを設定した場合の注意書きを追加
	8	タイトル変更
	8	イベントの実行制限、マスコット差し替え、着せ替え表現、まとめ(旧項番 5.7～5.10)を項番 9 に移動
	8.4	2D マチキャラの場合のモータ設定の制限事項を追加
	8.6	遷移タイミングを修正
	8.6	「キーフレーム」の説明を追加
	9	イベント文字列、イベント状態リセット、隠しマスコットイベントの実現方法、イベントの優先順位、まとめ(旧項番 6.7～6.11)を削除
	9.2.4	アニメーションの競合(旧項番 6.6)を移動
	9.3	タイトルを「2D アニメーション」から「2D マチキャラのキャラクタ作成」に変更
	9.3.1	マチキャラ Ver.1.0 の場合の最大アニメーション時間の注意書きを追加
	9.4	キャラクタの動作設定を追加
	9.5	キャラクタの移動を追加
	9.7	着せ替え表現(旧項番 5.9)を移動

	9.7	着せ替え表現にバルーンの説明を追加
	9.8	マスコット差替え(旧項番 5.8)を移動
	9.9	まとめ(旧 5.10)を移動
	9.9	ボーンの衝突、フレームレートの説明を追加
	9.9	ターゲット VRAM エリア、マスコットエリアの説明を追加
	12.1	タイトルを「ビヘビビア」から「ビヘビビアファイル」に変更
	12.1.1	ビヘビビアの設定方法の「dress」およびイベントを変更
	12.1.4	環境マッピングファイルについての注意書きを追加
	12.1.7	dress コマンドの「season」および「weekday」の設定値を追加
	12.1.7	ターゲット VRAM エリアの説明を追加
	12.1.8	イベントの種類、イベントコード仕様を追加
	12.1.8	同一コマンドが複数定義された場合の注意書きを追加
	12.1.9	useful コマンドの<id>に「0」が設定された場合の注意書きを追加
	12.1.9	前景／背景での画像ファイルの属性は全て「zoom」とする旨を追加
	12.1.9	fix コマンドの fit-mode、move-mode の設定値を追加 areasize コマンドの size の設定値を追加 useful コマンドの id の設定値を追加 scene コマンドの scene-no の設定値を追加
	12.1.9	areasize コマンドの設定値を変更
	12.1.9	マスコットエリアの説明を追加
	12.1.10	タイトルを「ビヘビビア使用例」から「ビヘビビアファイル設定例」に変更
	12.2	タイトルを「モータ」から「モータファイル」に変更
	12.2.2	前景／背景での画像ファイルの属性は全て「zoom」とする旨を追加
	12.2.3	タイトルを「モータ使用例」から「モータファイル設定例」に変更
	12.3.2	dirstr セクションを追加
	12.3.6	useful の書式および設定値を変更
	12.3.6	mask-col の色指定形式を追加
	12.3.6	cg の設定値を追加
	12.3.8	タイトルを「マチキャラメイクファイル使用例」から「マチキャラメイクファイル設定例」に変更
	12.5	定型移動データファイルを追加
	12.1.8	同一コマンドが複数定義された場合の注意書きを追加
	12.1.9	useful コマンドの<id>に「0」が設定された場合の注意書きを追加
3.0	全体	マチキャラ V3.0 の情報を追加
	2	構成を変更し、マチキャラメタデータファイルの説明を追加



	4	i コンシェルイベント、ポップアップ表示について追加
	6.3	マチキャラ Ver による配信方法の違いを追加
	7	ポップアップデータを追加
	9.7	ポップアップ表示に関する説明を追加
	12.3.6	popup セクションを追加
	12.3.8	cfURL セクションを追加
	12.3.9	cfmtime セクションを追加
	12.6	ポップアップデータファイルについて追加
3.1	4	i コンシェルイベントに「i スケジュール」を追加
	7	差分画像を利用しているアニメーション GIF について記載を追加
	9.6	誤記修正
	9.6.1	文字列置換機能の説明を追加
	12.4.3	文字列置換機能について追加
3.2	3.	Z バッファ・パースペクティブコレクションを追加
	3	マチキャラ Ver.2.0 について誤記を修正
	4.	Z バッファ・パースペクティブコレクションについての記述を追加
	7.	Z バッファ方式に対する記述を追加
	9.10	マスコットエリアについて注意書きを追加
4.0	全体	マチキャラ Ver.4.0 を追加
	全体	誤記など修正
	–	スクリプトリファレンスの項番のふりなおし
	4.1	移動機トリガーの新規イベントを追加
	4.8	ネットワークアップデートについて追加
	7.	マチキャラメーカーによる制限に「マチキャラコンテンツサイズ (V4.0)」、「アラーム、ダイアログ、i コンシェル TOP ショートカットフォーカス、i コンシェル TOP イベント」、「マチキャラメーカー V3.3 以前との差異」を追加
	8.6	アラームイベント、ダイアログイベント、i コンシェル TOP ショートカットフォーカスイベント、i コンシェル TOP イベントを追加
	9.4.3.3	重畳シーンを追加
	9.7	アラームイベント、ダイアログイベント、i コンシェル TOP ショートカットフォーカスイベント、i コンシェル TOP イベントを追加
	9.10	アラームイベント、ダイアログイベント、i コンシェル TOP ショートカットフォーカスイベント、i コンシェル TOP イベントを追加
	S3.1.8	アラームイベント、ダイアログイベント、i コンシェル TOP ショートカットフォーカスイベント、i コンシェル TOP イベントを追加

	S3.1.9	重畳シーンを追加
	S3.1.10	アラームイベント、ダイアログイベント、iコンシェルTOPショートカットフォーカスイベント、iコンシェルTOPイベントを追加
	S3.3	contentid セクション、fileid セクション、mmdurladdress セクション、mmdurlquery セクションを追加
	S3.3.1	バージョン変更
	S3.3.7	contentid セクションを追加
	S3.3.8	fileid セクションを追加
	S3.3.9	mmdurladdress セクションを追加
	S3.3.10	mmdurlquery セクションを追加
	S3.6	アラームイベント、ダイアログイベント、iコンシェルTOPショートカットフォーカスイベント、iコンシェルTOPイベントを追加
	S3.6.1	アラームイベント、ダイアログイベント、iコンシェルTOPショートカットフォーカスイベント、iコンシェルTOPイベントを追加
	3.	誤記修正、及び説明の追加

## はじめに

本書ではマチキャラを開発する際のガイダンスを解説します。

## 概要

本書ではマチキャラを作成するための手順や、作成する際に必要な概念を説明しています。

※マチキャラコンテンツを作成の際には、別途 NTT ドコモより提供するマチキャラコンテンツ作成ツール「マチキャラメーカー」をご利用ください。また、「マチキャラメーカー」に同梱されている「マチキャラメーカー操作説明書」もご覧ください。

## 必要となる知識

マチキャラを開発するにあたり以下の知識を必要とします。

- 3D およびグラフィックスに関する知識  
3D のマチキャラを作成する場合は 3D グラフィックスに関する知識が必要です。  
基本的な 3D やグラフィックスに関する知識を前提としていますが、不十分な場合でも一定の理解は可能です。
- ビヘイビア・モータの概念に関する知識  
マチキャラメーカーでデータを作成する際に必須となる知識です。  
本書にて説明致します。
- Windows に関する基本的な操作  
本書では特に説明しませんので、Windows に関する操作については別途書籍等で確認してください。

# マチキャラコンテンツ開発者ガイド

ここではマチキャラコンテンツを作成するために必要な知識や注意しなければならない点について説明します。

## 1. マチキャラとは

携帯電話の画面上にキャラクタ(マスコット)を表示、動作させて演出を行うサービスの名称です。

ユーザはこのサービスを使用するにあたり、使用するデータを自分の携帯端末上にダウンロードして選択することにより、携帯端末上に表示し動作させることができます。

※待ち受け画面以外へのマチキャラの表示の可否は端末により異なります。

## 2. マチキャラの種類

### 2.1. マチキャラデータ

携帯端末上で動作させるマチキャラのデータには構成されるデータの種類によって2種類あります。一つめは3Dモデルによって表現されたキャラクタを表示する**3D マチキャラ**です。もう一つは3Dモデルを使用せず、写真や絵などの画像を利用して表示する**2D マチキャラ**です。マチキャラを作成する際に使用するツールであるマチキャラメーカでは、この両方のデータを作成することができます。

### 2.2. マチキャラを構成するファイル

#### ■ マチキャラコンテンツファイル(cfd ファイル)

マチキャラコンテンツ本体のファイルです。マチキャラメーカで作成できます。

#### ■ マチキャラメタデータファイル(mmd ファイル)

マチキャラメタデータファイルです。マチキャラ Ver.3.0 以降のマチキャラコンテンツファイル(cfd ファイル)をダウンロードするために必要となります(マチキャラ Ver.1.0、2.0 では不要です)。マチキャラメーカで作成できます。

マチキャラメタデータファイルのフォーマットは以下となります。

要素	説明	プロパティ	値
バージョン番号	マチキャラメタデータファイルのバージョン番号	version	「1.0」の 3byte 固定
cfdURL	マチキャラコンテンツファイル(cfd ファイル)のダウンロード URL	dlurl	512byte 以内
日時情報	マチキャラコンテンツファイルの更新日時情報	time	YYYY/MM/DD/hh/mm/ss の 19byte 固定
マチキャラコンテンツファイルサイズ	マチキャラコンテンツファイルのファイルサイズ	filesize	最大 8byte 指定される数字は byte 単位

それぞれの項目は「プロパティ<sp>=<sp>値」となり、「<CR><LF>」にて区切られます

(<sp>は半角スペース 1byte)。プロパティは小文字とし、値は Shift-JIS 形式の ASCII 文字(または数字)となります。

#### <記述例>

```
version<sp>=<sp>1.0<CR><LF>
dlurl<sp>=<sp>http://www.XXX.co.jp/machichara/mascot.cfd<CR><LF>
time<sp>=<sp>2008/08/01/00/00/00<CR><LF>
filesize<sp>=<sp>2097152
```

### 3. マチキャラのバージョン

マチキャラではバージョンによって利用できる機能が異なります。バージョンによる違いは以下のとおりとなります。なお携帯電話が対応しているバージョンについては、別途提供する各携帯電話の情報をご参照ください。

#### ■ マチキャラ Ver.1.0

マチキャラの基本的な設定が可能なバージョンです。

マチキャラ Ver.1.0 対応機種では、マチキャラ Ver.1.0 のコンテンツを利用することができます。

#### ■ マチキャラ Ver.2.0

マチキャラ Ver.1.0 の機能に加えて、以下のイベントや機能が利用できます。

- ◇ 通話時間イベント
- ◇ 未読メールありイベント
- ◇ 積算通話時間、メール送信数によるマスコットの差し替え
- ◇ バルーン表示
- ◇ マスコットエリアの L、LL サイズ

マチキャラ Ver.2.0 対応機種では、マチキャラ Ver.1.0、2.0 のコンテンツを利用することができます。

#### ■ マチキャラ Ver.3.0

マチキャラ Ver.2.0 の機能に加えて、以下のイベントや機能が利用できます。

- ◇ i コンシェルイベント
- ◇ ポップアップ表示
- ◇ Z バッファ・パースペクティブコレクション(対応機種のみ)

マチキャラ Ver.3.0 対応機種では、マチキャラ Ver.3.0 のコンテンツを利用することができます。

※ Z バッファ・パースペクティブコレクション機能を総称して MascotCapsule V3 拡張機能と呼びます。

#### ■ マチキャラ Ver.4.0

マチキャラ Ver.3.0 の機能に加えて、以下のイベントや機能が利用できます。

- ◇ アラームイベント
- ◇ ダイアログイベント
- ◇ i コンシェル TOP ショートカットフォーカスイベント
- ◇ i コンシェル TOP イベント
- ◇ ネットワークアップデート

マチキャラ Ver.4.0 対応機種では、マチキャラ Ver.3.0、4.0 のコンテンツを利用することができます。

Ver.3.0 以降、ダウンロード時にマチキャラメタデータファイル(mmd ファイル)が必要となります。

それぞれのイベント、機能については以降の内容を参照してください。

## 4. マチキャラでできる演出

### 4.1. 各種イベントによるアクションの実行

不在着信、メール受信等が発生した際にキャラクタが特定の動作をするように設定することができます。設定できるイベントには、その場で動作する登場、定位置、癖イベントと、キャラクタが携帯端末画面上を移動する移動イベントがあります。

また、トリガーとなるイベントには以下の2種類があります。

#### ◇ 移動機トリガー

不在着信、メール受信、インフォメーション受信等、携帯端末で発生した事象がトリガーとなるイベントです。他に未読メールあり、通話時間、メーカイベントがあります。

インフォメーション受信時などに発行される、i コンシェルに関するイベントのことを i コンシェルイベントと呼びます。指定されたアクション種別により、以下の i コンシェルイベントが発行されます。

i コンシェルイベント	説明	発行元
デフォルト	アクション種別が「デフォルト」のインフォメーションを受信した場合、またはインフォメーションに指定されたアクション種別の i コンシェルイベント、アラームイベント、ダイアログイベント、i コンシェル TOP ショートカットフォーカスイベント、i コンシェル TOP イベントがマチキャラに設定されていない場合に発行されます。	ドコモ、IP
ドコモからのお知らせ	ドコモからのお知らせを受信した場合に発行されます。	ドコモ
ドコモのお知らせ(誕生日)	ドコモのお知らせ(誕生日)を受信した場合に発行されます。	ドコモ
鉄道運行情報	鉄道運行情報を受信した場合に発行されます。	ドコモ
道路交通情報	道路交通情報を受信した場合に発行されます。	ドコモ
気象・台風情報	気象・台風情報を受信した場合に発行されます。	ドコモ
地震情報	地震情報を受信した場合に発行されます。	ドコモ
地域イベント情報	地域イベント情報を受信した場合に発行されます。	ドコモ
自治体・生活関連情報	自治体・生活関連情報を受信した場合に発行されます。	ドコモ
笑う	アクション種別が「笑う」のインフォメーションを受信した場合に発行されます。	ドコモ、IP
驚く	アクション種別が「驚く」のインフォメーションを受信した場合に発行されます。	ドコモ、IP
悲しむ	アクション種別が「悲しむ」のインフォメーションを受信した場合に発行されます。	ドコモ、IP

挨拶する	アクション種別が「挨拶する」のインフォメーションを受信した場合に発行されます。	ドコモ、IP
謝る	アクション種別が「謝る」のインフォメーションを受信した場合に発行されます。	ドコモ、IP
i スケジュール	i スケジュールの自動更新が行われた場合に発行されます。	ドコモ※1
トルカ	トルカの自動更新が行われた場合に発行されます。	ドコモ※1
電話帳	電話帳の自動更新が行われた場合に発行されます。	ドコモ
今日の予定メッセージ ※2	今日の予定が表示される場合に発行されます。	携帯端末

※1 これらの i コンシェルイベントについては、IP(またはドコモ)が i スケジュールまたはトルカを更新することにより発行されますが、実際に更新が行われるタイミングは IP が指定できないため、発行元をドコモとしています。

※2 マチキャラ Ver.4.0 以降の機種で対応しています。

スケジュールに登録したアラームが鳴動する時に発行されるイベントをアラームイベントと呼びます。

音声対話機能を実行中に発行されるイベントをダイアログイベントと呼びます。ダイアログイベントには以下の種類があります。

ダイアログイベント	説明
音声対話(起動)	音声対話モード起動時に発行されます。自動的に「音声対話(発話起動)」に遷移します。
音声対話(発話起動)	音声対話モードの準備中に発行されます。自動的に「音声対話(発話待ち)」に遷移します。
音声対話(発話待ち)	音声対話モードの音声受付中に発行されます。
音声対話(認識終了)	音声対話モードで認識が完了した時に発行されます。
音声対話(エラー)	音声対話モードで認識失敗した時に発行されます。
音声対話(認識中)	音声対話モードで認識処理をしているときに発行されます。自動的に「音声対話(認識終了)」または「音声対話(エラー)」に遷移します。

待ち受け画面から i コンシェル TOP へ遷移するための i コンシェルアイコンをフォーカスした時に発行されるイベントを i コンシェル TOP ショートカットフォーカスイベントと呼びます。

i コンシェル TOP 表示画面で発行されるイベントを i コンシェル TOP イベントと呼びます。i コンシェル TOP イベントには以下の種類があります。

i コンシェル TOP イベント	説明
i コンシェル TOP	i コンシェル TOP が表示された場合に発行されます。
i コンシェル TOP(今日の予定)	i コンシェル TOP から今日の予定を表示した場合に発行されます。

#### ◇ スケジュール

携帯端末に登録されているスケジュールの日時になった時点で発生するイ

イベントです。スケジュールの種類ごとにイベントを設定できます。

#### 4.2. マスコットの差し替え

年月日、使用時間、積算通話時間、メール送信数およびメール受信数により使用するキャラクタを変更するように設定することができます。これにより、長く使用するとキャラクタが成長してゆくように演出することができます。

#### 4.3. マスコットの着せ替え

年、月、日、季節、曜日、時刻の組み合わせでキャラクタの外見を変化させることができます。これにより、季節や昼夜によって服を着せ替えるような演出をすることができます。3D マチキャラのみ設定することができます。

#### 4.4. バルーン表示

キャラクタの他にバルーンを表示することができます。バルーンは画像とテキストを設定することができ、キャラクタの吹き出しとして使用することができます。また、画像のみまたは画像とテキストどちらも表示しない設定もできます。

#### 4.5. ポップアップ表示

i コンシェルイベント、アラームイベント、ダイアログイベント、i コンシェル TOP ショートカットフォーカスイベント発行時に限り、キャラクタの他にポップアップを表示します。ポップアップの表示タイミングおよびポップアップに表示するテキストは移動機により指定されます。

#### 4.6. Z バッファ・パースペクティブコレクションでの描画

Z バッファ・パースペクティブコレクション機能でマスコットを描画します。(対応機種のみ。非対応機種では Z ソートで描画します。)

・Z ソート方式とは

3D における陰面処理のひとつです。視点から遠い距離にあるポリゴンから描画していき、隠れるポリゴンを消去する方式です。

・Z バッファ方式とは

3D における陰面処理のひとつです。ピクセルごとに奥行き方向の情報を記録して描画する方式です。

Z ソートと異なる点としてポリゴンが重なった状態でもどちらのポリゴンが手前にあるのか計算できる為、ポリゴンのちらつきが起こりにくくなっています。

・パースペクティブコレクション機能とは

投影方法が透視投影の時にポリゴンに張ったテクスチャが歪む現象を補正する機能です。

#### 4.7. 3D に特化した演出

キャラクタ自体の演出としては、平行光源、透視投影、環境マッピング等を設定することができます。平行光源、透視投影および環境マッピングは 3D のみ設定することができます。

#### 4.8. ネットワークアップデート

ユーザがダウンロード、保存したマチキャラを IP サーバにアップロードした新しいマチキャラに更新することができます。

##### 4.8.1. CFDID

ネットワークアップデートするマチキャラを特定するため、マチキャラに CFDID を設定する必要があります。

CFDID とは、CFD コンテンツ ID (32 桁) と CFD ファイル ID (32 桁) の 2 つから構成される



64 桁の ID です。CFDID に利用できる文字は英字大文字・小文字及び数字の 62 種類です。

- ・ CFD コンテンツ ID : 00000000000000000000000000000000
- ・ CFD ファイル ID : abcdefghijklmnopqrstuvwxyz123456

それぞれの値を上記とした場合の CFDID は

「00000000000000000000000000000000abcdefghijklmnopqrstuvwxyz123456」

となります。

◇ CFD コンテンツ ID

CFD コンテンツ ID は「00000000000000000000000000000000」固定です。

◇ CFD ファイル ID

CFD ファイル ID とはファイルの種類を判別することを目的とした 32 桁の ID です。IP により任意に指定することができます。

#### 4.8.2. ネットワークアップデート動作概要

ネットワークアップデートを利用して、ユーザ操作でマチキャラを更新することができます。ネットワークアップデートを実施できるマチキャラは、マチキャラ Ver.4.0 以降の CFDID、MMD ファイル取得先 URL が正しく設定されているマチキャラに限ります。

#### 4.8.3. 更新契機

以下の契機により、ネットワークアップデートをすることができます。

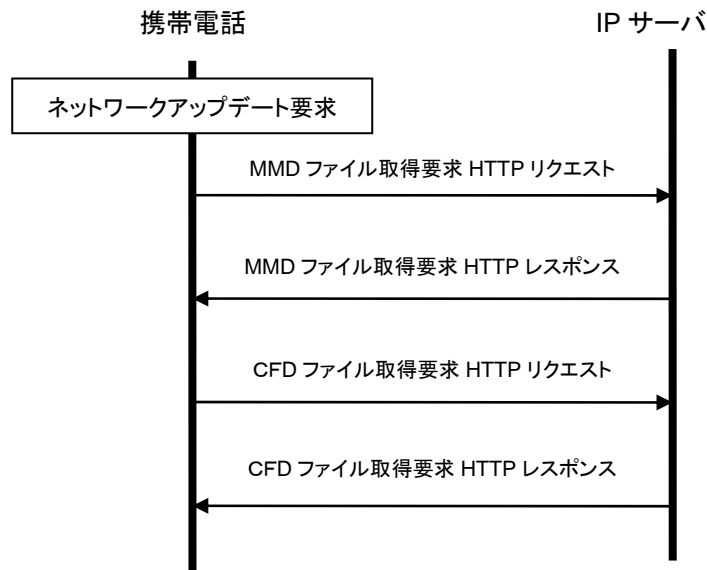
- ・ ユーザ操作による手動更新（マチキャラのサブメニューから更新実施）

◇ ユーザ操作による手動更新

ユーザがマチキャラを表示し、サブメニューから「手動アップデート」を選択することにより、ネットワークアップデートを実行できます。

#### 4.8.4. ネットワークアップデート動作概要

ネットワークアップデートでは以下の図のような処理が行われます。



#### ①MMD ファイル取得要求 HTTP リクエスト

ネットワークアップデート要求があった際に、携帯電話は MMD ファイル取得要求 HTTP リクエストを送信します。この時の URL は、更新対象となる CFD ファイル内に設定された MMD ファイル URL (アドレス)、MMD ファイル URL (クエリ)、及び CFDID となります。

例)

- ・ MMD ファイル URL (アドレス) : <http://docomo.ne.jp/machichara/sample.mmd>
- ・ MMD ファイル URL (クエリ) : id=0001&test=on
- ・ CFD コンテンツ ID : 00000000000000000000000000000000
- ・ CFD ファイル ID : cccccccccccccccccccccccccccccccccccc

とした場合のリクエスト URL は

<http://www.nttdocomo.co.jp/machichara/sample.mmd?id=0001&test=on&cfdid=00000000000000000000000000000000ccccccccccccccccccccccccccccccccccc>

となります。

#### ②MMD ファイル取得要求 HTTP レスポンス

HTTP ヘッダに「X-IB-MCHARA-CTRCODE」ヘッダをつけて MMD ファイルを応答します。

「X-IB-MCHARA-CTRCODE」には以下の値を設定します。

- ・ 200 : ダウンロード対象ファイルを正常に送信
- ・ 304 : 更新の必要なし
- ・ 500 : 何らかの要因で更新ができない

例)

```

HTTP/1.1 200 OK<CR><LF>
Data: Thu, xx Nov 201x 09:00:00 GMT<CR><LF>
X-IB-MCHARA-CTRCODE: 200<CR><LF>
<CR><LF>
(送信するファイルの内容)
  
```

#### ③CFD ファイル取得要求 HTTP リクエスト

MMD ファイルのダウンロードが完了した場合、携帯電話は CFD ファイル取得要求 HTTP リクエストを送信します。この時の URL は取得した MMD ファイル内に設定された cfdURL、及び CFDID となります。

例)

- ・ cfdURL : <http://docomo.ne.jp/machichara/sample.cfd?id=0001&test=on>
- ・ CFD コンテンツ ID : 00000000000000000000000000000000
- ・ CFD ファイル ID : ccccccccccccccccccccccccccccccccccc

とした場合のリクエスト URL は

<http://www.nttdocomo.co.jp/machichara/sample.cfd?id=0001&test=on&cfdid=00000000000000000000000000000000cccccccccccccccccccccccccccccccccc>

となります。

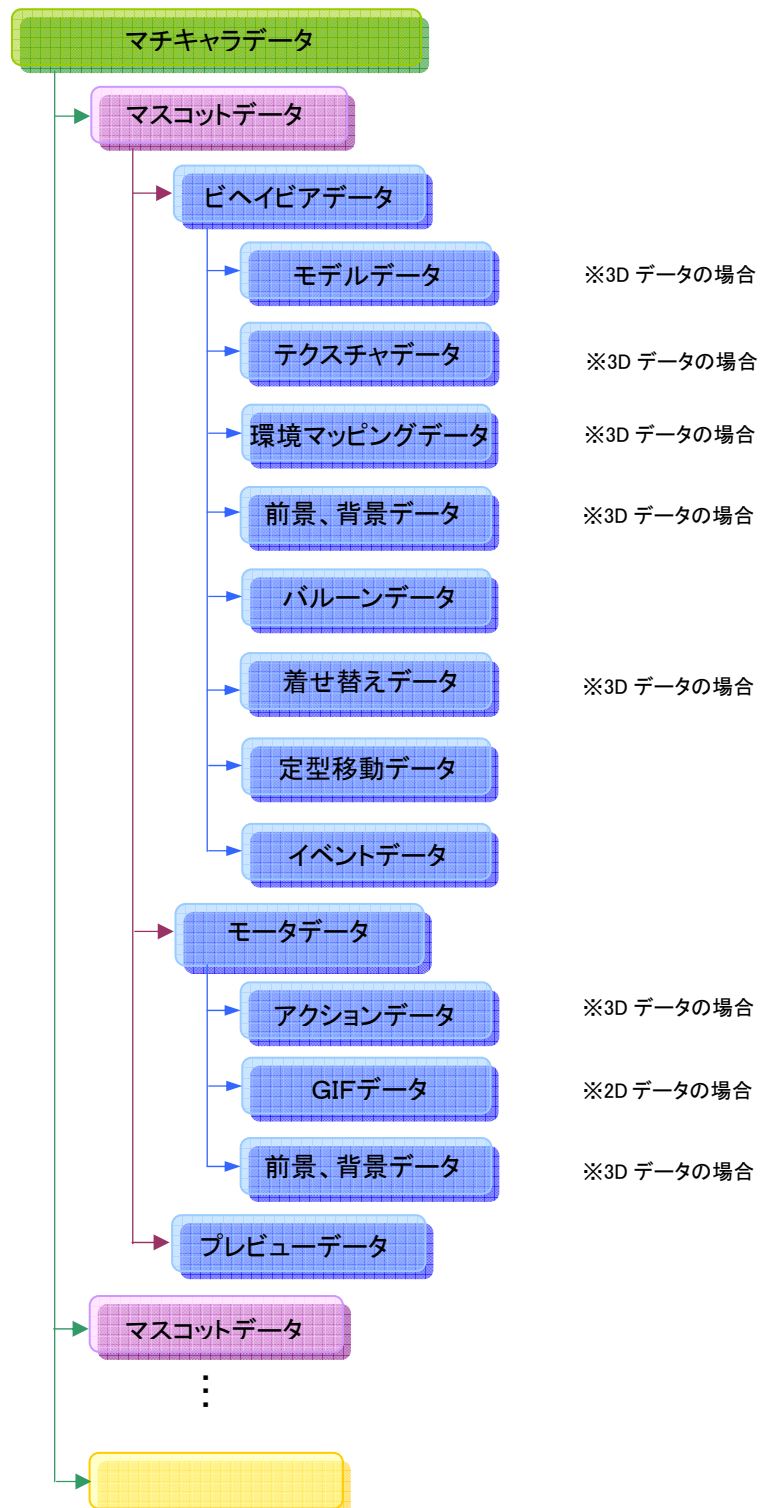
#### ④CFD ファイル取得要求 HTTP レスポンス

HTTP ヘッダに「X-iB-MCHARA-CTRCODE」ヘッダをつけて CFD ファイルを応答します。

「X-iB-MCHARA-CTRCODE」の設定規則は「②MMD ファイル取得要求 HTTP レスポンス」と同じです。

## 5. マチキャラデータの構成

マチキャラデータは以下の要素から構成されています。

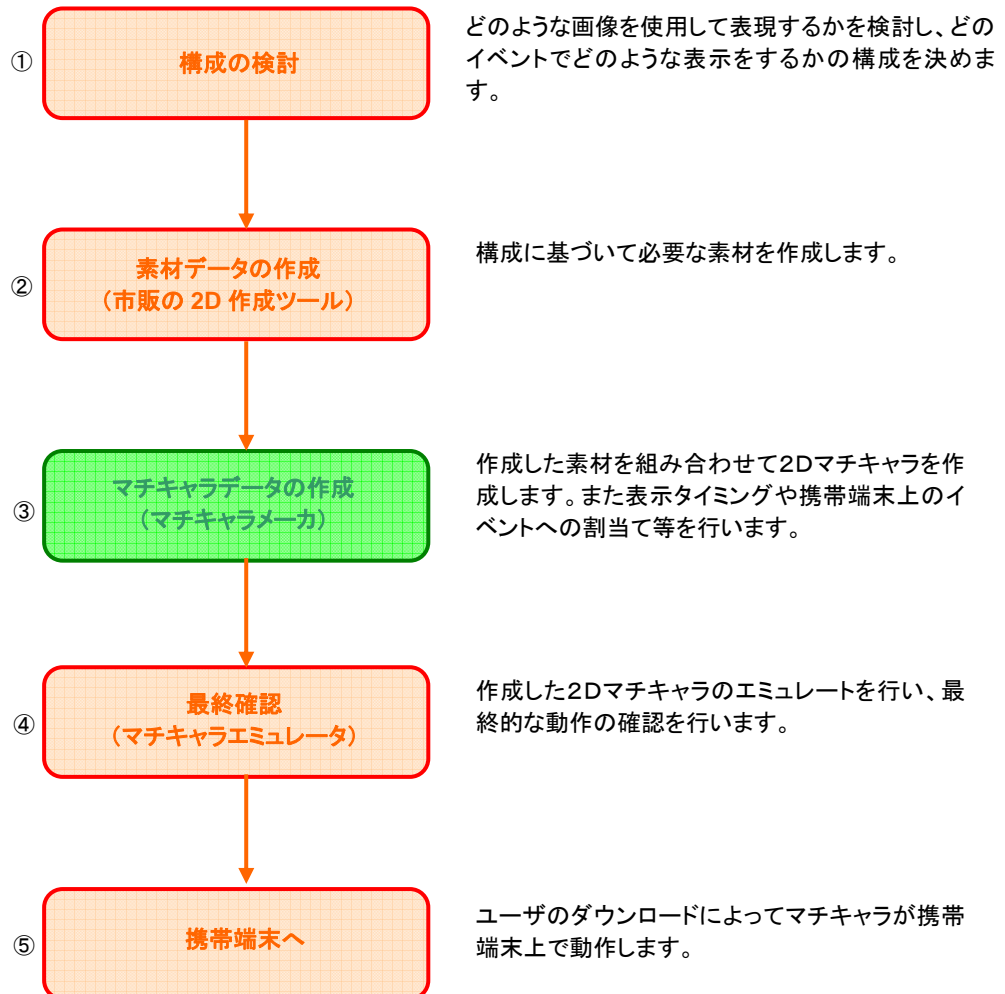


## 6. マチキャラコンテンツ作成の流れ

マチキャラコンテンツを作成するための大まかな流れを示します。コンテンツ作成の流れは作成するマチキャラの種類によって以下の2つに分かれます。

### 6.1. 2D マチキャラの場合

2D マチキャラの作成手順は以下のような流れになります。



### ① 構成の検討

2D マチキャラでは複数枚の画像を切り替えることによって動いているように見せるため、表現を行うための画像が必要になります。これらを各イベントに割当て、そのイベントが発行された時の動作として表現されます。2D マチキャラの構成を検討する場合には後述する「ガイダンス」に従って検討を行ってください。

✓ 使用可能な画像の枚数には制限があります。

### ② 素材データの作成

検討した構成に基づいて画像データを作成します。作成するツールは市販の 2D 作成ツールで作成を行います。

✓ 画像は指定されたフォーマットである必要があります。

### ③ マチキャラデータの作成

作成した素材を用いてマチキャラを作成するツールであるマチキャラメーカーによりデータを作成します。マチキャラメーカーにはプレビュー機能がありますので、ある程度のデータの確認はできるようになっています。マチキャラメーカーの使用方法については「マチキャラメーカー操作説明書」を参照してください。

### ④ 最終確認

マチキャラエミュレータを用いて最終的なコンテンツの確認を行います。ここでは携帯端末とほぼ同じ環境で動作確認を行うことができます。エミュレータ上での確認が終了するとコンテンツの開発は終了となります。

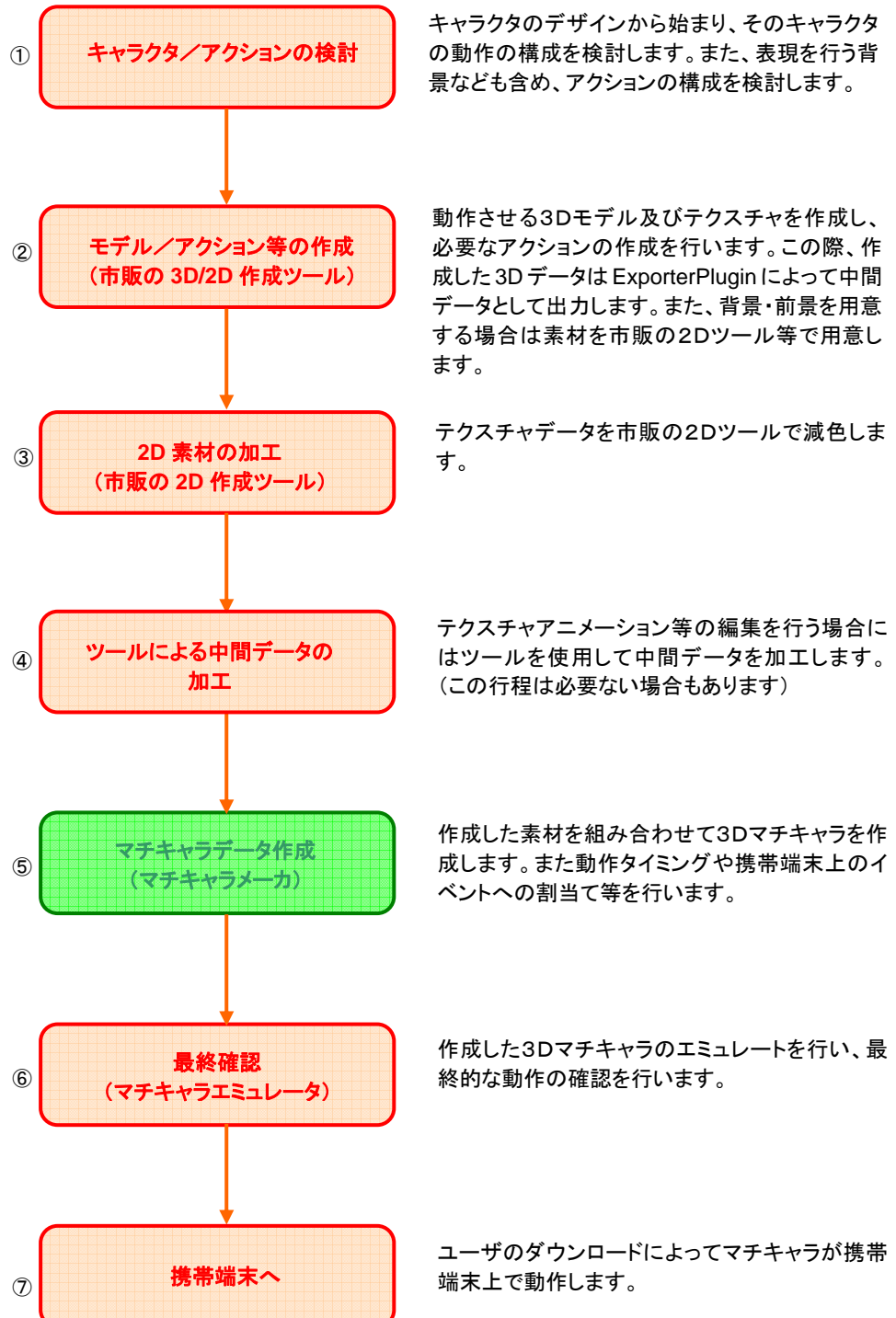
### ⑤ 携帯端末へ

実際に携帯端末上にマチキャラをダウンロードして操作します。

✓ マチキャラデータで設定する色はRGB888形式で作成しますが、実際の携帯端末はRGB565形式である場合があります。その場合は、設定した色コードの近似色が使用されることになります。

## 6.2. 3D マチキャラの場合

3D マチキャラの作成手順は以下のような流れになります。



**① キャラクタ／アクションの検討**

キャラクタのイメージを作成し、そのキャラクタにどのような動作をさせるのかを検討します。この際、マチキャラを作成するために必要な「ビヘイビア／モータの概念」をある程度意識しておく必要があります。この段階ではキャラクタの動きのパターンやこの概念に基づいてイベントに割当ててアクションを検討します。

✓ 表現できるアクションの数には制限があります。

**② モデル／アクション等の作成**

検討した構成に基づいて 3D モデルやテクスチャ、アクションを作成します。3D モデルを作成する際の注意点や制限などは後述の「ガイダンス」に基づいている必要があります。作成したデータは ExporterPlugin によって出力し、中間データとしておく必要があります。

✓ 各種 3D ツールにアドインする ExporterPlugin については該当のマニュアルを参照してください。

✓ 使用可能なアクション数やモデルの属性については後述の「マチキャラコンテンツ作成における制限事項」を参照してください。

**③ 2D 素材の加工**

テクスチャデータを減色します。

BMP であれば 8bit256 色 Max256x256、GIF であれば GIF89a 形式アニメーション GIF 以外 Max256x256 にします。

**④ ツールによる中間データの加工**

必要であれば ExporterPlugin によって出力された中間データの微調整を行います。ExporterPlugin については別途マニュアルを参照してください。

**⑤ マチキャラデータの作成**

作成した素材を用いてマチキャラメーカーによりデータを作成します。

マチキャラメーカーにはプレビュー機能がありますのである程度のデータの確認はできるようになっています。マチキャラメーカーの使用方法については「マチキャラメーカー操作説明書」で解説していますので参照してください。

**⑥ 最終確認**

マチキャラエミュレータを用いて最終的なコンテンツの確認を行います。ここでは移動機とほぼ同じ環境で動作確認を行うことが可能です。エミュレータ上での確認が終了するとコンテンツの開発は終了となります。

**⑦ 携帯端末へ**

実際に携帯端末上にマチキャラをダウンロードして操作します。

✓ マチキャラデータで設定する色は RGB888 形式で作成しますが、実際の携帯端末は RGB565 形式である場合があります。その場合は、設定した色コードの近似色が使用されることになります。



### 6.3. コンテンツの配信

マチキャラコンテンツの配信方法について説明します。

また、マチキャラコンテンツは RangeRequest 機能に対応しているため、コンテンツの取得が中断された場合も、続きから再取得することが出来ます。RangeRequest 機能に対応する場合の注意点についても説明します。

#### 6.3.1. マチキャラ Ver.1.0、2.0 の場合

マチキャラ Ver.1.0、2.0 の配信方法は以下となります。配信時はマチキャラコンテンツファイル(cfd ファイル)を直接指定します。

##### (1) <A>タグでのコンテンツ配信

マチキャラのコンテンツは、HTML の<A>タグを用いてリンクを設定します。

##### <記述例>

```
<A href="http://www.hogehoge.co.jp/mascot.cfd">マチキャラ君</A>
```

※ マチキャラデータを配信する Web サーバでは、マチキャラファイルの Content-Type が “application/x-character-overlay” に設定されている必要があります。

※ マチキャラデータを CGI 経由でダウンロードする場合は必ず Content-Length を付与してください。

##### (2) <OBJECT>タグでのコンテンツ配信

以下に示すように<OBJECT>タグを利用してもマチキャラファイルを配信することが可能です。

##### <マチキャラコンテンツ配信に必要な要素と属性一覧>

要素	属性	属性値	備考
OBJECT	data	マチキャラコンテンツファイル(cfd ファイル)の URI	-
	type	application/x-character-overlay	マチキャラデータであることを示す識別子
	id <sup>※1</sup>	任意の識別子	OBJECT を特定するための識別子
	declare	-	オブジェクト宣言であることを示す識別子
A	href	参照する OBJECT ID	OBJECT の ID を指定

※1 id 属性の値は半角英字で始まり、半角英数字、“-”(ハイフン)、“\_”(アンダースコア)、“.”(コロン)、“.”(ピリオド)で構成されている必要があります。

##### <記述例>

```
<OBJECT declare id="mascot" data="mascot.cfd" type="application/x-character-overlay">
</OBJECT>
<A href="#mascot">マチキャラ君</A>
```

#### 6.3.2. マチキャラ Ver.3.0 以降の場合

マチキャラ Ver.3.0 以降の配信方法は以下となります。配信時はマチキャラメタデータファイル(mmd ファイル)を指定します。

**<OBJECT>タグでのコンテンツ配信**

以下に示すように<OBJECT>タグを利用してマチキャラファイルを配信することが可能です。

**<マチキャラコンテンツ配信に必要な要素と属性一覧>**

要素	属性	属性値	備考
OBJECT	data	マチキャラメタデータファイル (mmd ファイル) の URI	–
	type	application/x-character-overlay	マチキャラデータであることを示す識別子
	id※1	任意の識別子	OBJECT を特定するための識別子
	declare	–	オブジェクト宣言であることを示す識別子
A	href	参照する OBJECT ID	OBJECT の ID を指定

※1 id 属性の値は半角英字で始まり、半角英数字、“-” (ハイフン)、“\_” (アンダースコア)、“.” (コロン)、“.” (ピリオド) で構成されている必要があります。

**<記述例>**

```
<OBJECT declare id="mascot" data="mascot.mmd" type=" application/x-character-overlay ">
</OBJECT>
<A href="#mascot">マチキャラ君</A>
```

**6.3.3. RangeRequest 機能の HTTP リクエストレスポンス**

RangeRequest 機能を用いて、取得途中のマチキャラコンテンツを途中から取得する為に、端末から If-Unmodified-Since ヘッダ及び Range ヘッダが送出されます。これに対して、Web サーバからは上記リクエストに適切なレスポンスとして、Last-Modified ヘッダ、Accept-Ranges ヘッダ、Content-Type ヘッダ、Content-Range ヘッダを送出する必要があります。以下に HTTP リクエストとレスポンスの例について記載します。

(例)コンテンツのサイズが 331977byte のマチキャラコンテンツを取得する。

**1: 最初のマチキャラコンテンツデータリクエスト例**

```
GET /dir/mascot.cfd HTTP/1.1
Host: xxx.xxx.xxx
User-Agent: DoCoMo/2.0 ●90xi●(c●●●●;TB;W●H●)
```

**2: 通常応答例**

```
HTTP/1.1 200 OK
Date: Fri, 01 May 2006 08:24:57 GMT
Last-Modified: Fri, 11 Oct 2002 03:32:29 GMT
Accept-Ranges: bytes
Content-Type: application/x-character-overlay
Content-Length: 331977
(マチキャラコンテンツデータ)
```

(例)コンテンツのサイズが 331977byte のマチキャラコンテンツを取得した際、330735byte まで取得し、続きを取得する。

### 3.部分コンテンツリクエスト例

```
GET /dir/mascot.cfd HTTP/1.1
Host: xxx.xxx.xxx
User-Agent: DoCoMo/2.0 ●90xi●(c●●●●;TB;W●H●)
If-Unmodified-Since: Fri, 11 Oct 2002 03:32:29 GMT
Range: bytes=330735-331976
```

#### ※注意

RangeRequest ではコンテンツのサイズを 0 からカウントする為、部分リクエストの range では 1 バイト少ない要求が送出されます。

### 4.部分応答例

```
HTTP/1.1 206 Partial Content
Date: Fri, 01 May 2006 08:24:58 GMT
Last-Modified: Fri, 11 Oct 2002 03:32:29 GMT
Accept-Ranges: bytes
Content-Type: application/x-character-overlay
Content-Range: bytes 330735-331976/331977
Content-Length: 1242
(マチキャラコンテンツデータ)
```

#### ※注意

Content-Range には  
(取得したコンテンツサイズ)-(リクエストで要求されたコンテンツサイズ)/実サイズ  
Content-Length には  
(リクエストで要求されたコンテンツサイズ)-(取得したコンテンツサイズ)+1 バイト  
を設定してください。

### 5:If-Unmodified-Since の条件が不適合の場合

```
HTTP/1.1 412 Precondition Failed
Date: Fri, 01 Apr 2005 08:24:58 GMT
Content-Type: text/plain
Content-Length: 9

Precondition Failed
```

#### RangeRequest 機能に対応される際の注意点(重要)

- RangeRequest 対応端末においては、IP サーバからのレスポンスヘッダに Accept-Range ヘッダが無い場合は、マチキャラコンテンツの取得ができませんのでご注意ください。
- CGI を利用してマチキャラコンテンツを配信される際にも、必ず Last-Modified ヘッダ、Accept-Ranges ヘッダ、Content-Type ヘッダ、Content-Range を送出する設定にしていた

だく必要があります。

- ・ WEB サーバの各種設定については各サーバのマニュアルをご参照ください。

#### 6.4. マチキャラコンテンツの DRM

マチキャラコンテンツはコンテンツ内の再配付不可識別子にかかわらず、一律再配布不可となります。

## 7. マチキャラコンテンツ作成における制限事項

ここではマチキャラコンテンツを作成する上での制限事項を示します。

### ◆ 使用可能な 3D ツール

3D マチキャラを作成する際の 3D ツールは以下のいずれかである必要があります。

1. 3DStudio Max
2. XSI
3. LightWave
4. MAYA

また、作成したデータをマチキャラメーカーで使用するためには、上記各ソフトウェアに対応した ExporterPlugin を利用する必要があります。

ExporterPlugin および使用方法等については、株式会社エイチアイのホームページ (<http://www.mascotcapsule.com/toolkit/docomo/ja/index.html>) を参照してください。

※マチキャラでは MascotCapsule V3 を使用しています。

### ◆ 中間フォーマットデータ

各 3D ツールの ExporterPlugin によって出力される中間フォーマットデータには以下の種類があります。

#### 1. BAC6(モデル)/ TRA4(アクション)フォーマット

最新の ExporterPlugin によって出力された中間ファイルフォーマットです。作成した 3D データをマチキャラメーカーで使用するためには、この BAC6/TRA4 形式のデータフォーマットである必要があります。

#### 2. BAC5(モデル)/ TRA3(アクション)フォーマット

別途ツール(PAC.exe)を使用することにより BAC6(モデル)/ TRA4(アクション)フォーマットに変換することができます。作成した 3D データをマチキャラメーカーで使用するためには、BAC6/TRA4 形式に変換する必要があります。

### ◆ 作成する 3Dモデル

3Dモデルを作成する際には以下のような点に注意する必要があります。

#### 1. ポリゴン数

ポリゴンデータ作成時にポリゴン数を増やすと、それだけプログラムで展開した時に必要とされるメモリが多くなります。メモリに展開できればポリゴン数に関係無く表示可能ですが、携帯端末上であることを意識して、ポリゴン数は **1000** ポリゴン程度を推奨します。

以下に参考値としてモデル作成時の各ポリゴン数でのデータの目安を示します。

ポリゴン数	必要メモリ試算(Byte)	データサイズ約(Byte)
500	8,200	4,000
750	12,000	8,000
1,000	14,700	8,000
2,000	30,000	19,000
3,000	46,000	31,000
5,000	73,400	43,000
10,000	148,000	91,000

また、マチキャラにおける描画方式はポリゴン単位の Z ソート方式とピクセル単位の Z バッファ方式を使用しています。

どちらの方式が使用されているかは携帯端末機種によって違います。

Z ソート方式に対応する機種ではポリゴンが重なるとちらつくことがあります。  
Z バッファ方式に対応する機種ではポリゴンが重なるとちらつく現象が Z ソート方式の機種より軽減されます。  
モデリングを行う際にはそれぞれの方式の特性に合った方法で作成してください。

## 2. 使用可能なポリゴン属性

マチキャラを作成するために使用可能な 3D データのポリゴン属性は以下のとおりです。

属性		使用可／不可
片面		使用可
両面		使用可
透過		使用可
Multiple Texture		使用可
Flat Color Polygon		使用可
Texture Animation		使用可
ブレンディング	半透過	使用可
	加算	使用可
	減算	使用可
光源		使用可
環境マッピング		使用可

3D モデルの該当するポリゴン属性が無効に設定されていると、マチキャラデータに設定しても機能は反映されません。マチキャラで使用する機能は 3D モデルを作成する際に機能を有効にするようにポリゴン属性を設定しておく必要があります。

- ✓ ブレンディングを使用する際は注意が必要です。(後述する「9.10.まとめ」の「作成するモデルの表現」参照)

## 3. 画面への描画方法

画面への描画は**平行投影**または**透視投影**で描画されます。

## 4. テクスチャ

モデルに設定するテクスチャは必要であれば減色する必要があります。テクスチャの種類は BMP または GIF で以下の属性である必要があります。

種類	属性
BMP	8bit、256 色、無圧縮形式 サイズの上限は 256 x 256
GIF	GIF89a 形式、2～256 色、透過色指定あり GIF アニメーションは不可 サイズの上限は 256 x 256

透過は GIF 形式で透過色の指定があるもの、または BMP の場合は 0 番パレットが該当のポリゴン属性に透過を指定した場合に行われます。

GIF の属性としてインターレース属性を設定することは可能ですが、インターレース効果は反映されません。

テクスチャの使用によるメモリの使用量は、テクスチャ 1 枚につき約 80K を消費します。

## 5. アクションについて

マチキャラではアクションデータには以下の 2 種類があります。

通常アクション・・・マスコットのボーンアニメーション、テクスチャアニメーション(動的ポリゴン)などに利用します。

時間反映(着せ替え)・・・時間反映や隠しマスコットを目的として作成される外見変化用のデータ(動的ポリゴン)です。

作成したアクションを ExporterPlugin で出力するとアクションファイルにはモデルの全てのボーンが含まれている状態で出力されます。マチキャラでアクションを再生する際に不都合が生じる場合がありますので後述するマチキャラメカにて必要なボーンのみを選択する必要があります。

アクションの使用によるメモリの使用量は、1アクションにつきモデルの 1/5～1/10 程度消費します。

通常アクションを作成する際のフレームレートは 30fps で作成してください。モデルをパーツに分けて各パーツを同時に再生するようなアクションを作成する場合には各パーツのアクションの時間を同じにしてください。また、フルサイズイベントで使用するアクションファイルはアクションの時間を 20 ミリ秒単位で作成してください。

時間反映を作成する場合は動的ポリゴン操作のみを設定します。

#### ◆ 背景・前景データ

マチキャラで背景・前景データを使用する際は以下の点に注意してください。  
画像データの種類の BMP または GIF で以下の属性である必要があります。

種類	属性
BMP	8bit、256 色、無圧縮形式 サイズは Max 256x256、推奨 QCIF
GIF	GIF89a 形式、2～256 色 GIF アニメーションは不可 サイズは Max 256x256、推奨 QCIF

画像の使用によるメモリの使用量は、画像1枚につき約 80K を消費します。  
なお、背景データは透過しません。前景データは BMP の場合は 0 番パレットが、GIF の場合は指定した透過色が透過されます。また、透過なし GIF が前景に指定された場合は 0 番パレットが透過されます。  
GIF の属性としてインターレース属性を設定することは可能ですが、インターレース効果は反映されません。

#### ◆ 環境マッピングデータ

マチキャラで環境マッピングデータを使用する際は以下の点に注意してください。  
画像データの種類の BMP または GIF で以下の属性である必要があります。

種類	属性
BMP	8bit、256 色、無圧縮形式 サイズは Max 64x64
GIF	GIF89a 形式、2～256 色 GIF アニメーションは不可 サイズは Max 64x64

画像の使用によるメモリの使用量は、画像1枚につき約 80K を消費します。  
環境マッピングデータは該当のポリゴン属性に環境マッピングを指定した場合に行われます。GIF の属性としてインターレース属性を設定することは可能ですが、インターレース効果は反映されません。

#### ◆ プレビューデータ

マチキャラでプレビューデータを使用する際は以下の点に注意してください。

画像データの種類は BMP、GIF または JPG で以下の属性である必要があります。

種類	属性
BMP	8bit、256 色 サイズは 160x160
GIF	GIF89a 形式、2～256 色 サイズは 160x160
JPG	JFIF 形式、 サイズは 160x160

GIF の属性としてインターレース属性を設定することは可能ですが、インターレース効果は反映されません。

#### ◆ アニメーション GIF データ

2D のマチキャラで使用する GIF データは以下の点に注意してください。  
画像データの種類は GIF で以下の属性である必要があります。

種類	属性
GIF	GIF89a 形式、2～256 色 サイズは以下のいずれか 32x32、48x48、64x64、96x96、128x128、 192x192、256x256、288x288、384x384

GIF の属性としてインターレース属性を設定することは可能ですが、インターレース効果は反映されません。

差分画像を利用しているアニメーション GIF (差分最適化されたアニメーション GIF) を 2D マチキャラの画像データとして利用した場合、機種によって正常に表示できない場合があります。

#### ◆ バルーン(吹き出し)データ

マチキャラでバルーンデータを使用する際は以下の点に注意してください。  
画像データの種類は GIF のみで以下の属性である必要があります。

種類	属性
GIF	GIF89a 形式、2～256 色 GIF アニメーションは不可 サイズは以下のいずれか 48 × 16、96 × 32、192 × 64、288x96

GIF の属性としてインターレース属性を設定することは可能ですが、インターレース効果は反映されません。

#### ◆ ポップアップデータ

マチキャラでポップアップデータを使用する際は以下の点に注意してください。  
画像データの種類は GIF のみで以下の属性である必要があります。

種類	属性
GIF	GIF89a 形式、2～256 色 GIF アニメーションは不可 サイズは以下のいずれか 228 × 186、456 × 372、684x558 画像データサイズは 300K バイト以下とする

GIF の属性としてインターレース属性を設定することは可能ですが、インターレース効果は反映されません。



## ◆ メモリ概算

マチキャラを動作させるために必要な概算メモリは以下のようになります。

素材	使用メモリ(約)
モデル	20K(750 ポリゴンの場合) x モデル数
テクスチャ	80K(サイズに限らず) x テクスチャ数
環境マッピング	80K(サイズに限らず) x 環境マッピング数
アクション	(モデルの 1/10)K x アクション数
重畳表示エンジン	100K(エンジン本体 30K + GIF/BMP 変換用 66K)
3D エンジン	200K
バルーン	(112byte x バルーン画像数) + 描画処理分

上記の合計値が実際に必要になるメモリとなります。

プレビュー、アニメーション GIF およびポップアップを表示するために必要なメモリは携帯端末メーカーにより異なります。

## ◆ マチキャラメーカーによる制限

マチキャラコンテンツを作成するために使用するツール、マチキャラメーカーでは以下の制限を設けていますのでマチキャラを設計する際に注意してください。

項目	制限内容
マチキャラコンテンツサイズ (V1.0/V2.0)	500 Kバイト以内
マチキャラコンテンツサイズ (V3.0)	2M バイト以内
マチキャラコンテンツサイズ (V4.0)	5M バイト以内
マスコット数	64 個以内
ビヘイビア名の長さ	64 バイト以内
ビヘイビアファイル名の長さ	64 バイト以内
マスコット毎のテクスチャ総数 (背景・前景を含む)	16 ファイル以内
マスコット毎のアクションデータ総数	512 ファイル以内
マスコット毎の環境マッピングデータ数	1 ファイル以内
マスコット毎のプレビューデータ数	1 ファイル以内
マスコット ID	128 文字以内
マチキャラタイトル名	30 文字以内
マチキャラサブタイトル名	30 文字以内
マチキャラ更新先 URL	500 文字以内
マスコット毎のイベント数	256 イベント以内 (デフォルト状態定位置イベントは設定必須、V3.0 以降の場合は i コンシェル定位置イベントは設定必須)
隠しマスコットのイベント ID の長さ	14 文字
メーカーイベント ID の長さ	16 文字
i コンシェル、アラーム、ダイアログ、i コンシェル TOP ショートカットフォーカス、i コンシェル TOP イベント ID の長さ	9 文字
その他イベント ID の長さ	7 文字
モデルファイル名の長さ	64 バイト以内
テクスチャファイル名の長さ	64 バイト以内
環境マッピングファイル名の長さ	64 バイト以内
モータ名の長さ	64 バイト以内
モータファイル名の長さ	64 バイト以内
ステート名の長さ	16 バイト以内
マスコット毎のモータ数	3D 16 モータ以内 2D 1 モータ
モータ毎のステート	256 ステート以内
マスコット毎のモータステート数	512 ステート以内
マスコット毎のステートのファイル数	3D 1 ファイル以内 2D 移動以外 1 ファイル 2D アニメーションファイル数 移動 8 ファイル
モータステートのイベント数	16 イベント以内
モータステートのイベント毎のコマンド数	16 コマンド以内
アクションファイル名の長さ	64 バイト以内

アニメーションファイル名の長さ	64 バイト以内
マスコット毎の定型移動ファイル数	1 ファイル以内
定型移動データファイル名の長さ	64 バイト以内
定型移動ルート名の長さ	16 バイト以内
マスコット毎の定型移動ルート数	32 ルート以内
定型移動ルートポイント数	64 ポイント以内
マスコット毎のバルーンファイル数	1 ファイル以内
バルーンデータファイル名の長さ	64 バイト以内
マスコット毎のバルーンステート数	300 ステート以内
マスコット毎のバルーンテキスト数	2400 個以内
バルーン名毎のGIFデータ数	32 ファイル以内 左上、右上、右下、左下それぞれ 8 ファイル以内
バルーンテキスト設定カラー数	256 色以内
バルーンの表示間隔	最大 60000 ミリ秒
プレビューイメージファイル名の長さ	64 バイト以内
ポップアップデータファイル名の長さ	64 バイト以内
ポップアップイメージファイル名の長さ	64 バイト以内
ポップアップ名の長さ	16 バイト以内
ポップアップデータ数	1 データ以内(V3.0 以降では設定必須)
ポップアップデータ毎の画像数	3 ファイル以内

また、マチキャラメーカを使用する場合には以下の点に注意してください。

#### 1. マチキャラメーカでの編集

以下の項目がスクリプトで未設定の場合、マチキャラメーカで自動的に編集されます。スクリプトを作成する際には注意してください。(スクリプトについては、後述の「スクリプトリファレンス」を参照してください。)

- (1)マスコットのサブタイトル
- (2)モータ名

モータ名はスクリプトでは定義していません。マチキャラメーカで、モータファイルの先頭行のコメントをモータ名として扱います。コメントに文字数の制限はありませんが、マチキャラメーカで64文字のみモータ名とするため、マチキャラメーカに読み込んで保存した後は、先頭行は最大64文字となります。

#### 2. マチキャラメーカ V1.0 との差異

以下の項目はマチキャラメーカ V2.0 以降とマチキャラメーカ V1.0 では処理方法が異なります。マチキャラコンテンツを作成する際は注意してください。

##### (1)modeloffset コマンド

マチキャラメーカ V2.0 以降ではイベント毎に異なるモデルオフセットを設定することができません。全てのイベントで同じ値を設定する必要があります。

##### (2)modelscale コマンド

マチキャラメーカ V2.0 以降ではイベント毎に異なるモデルサイズを設定することができません。全てのイベントで同じ値を設定する必要があります。

##### (3)setfg コマンド

setfg コマンドを省略すると、マチキャラメーカ V2.0 以降では「前景を表示しない」と解釈します。直前に実行されていたイベントの前景を引き継ぐことはできません。イベント毎に setfg コマンドを設定してください。なお、前景を無効にするための setfg コマンドは設定する必要がなくなります。

#### (4)setbg コマンド

setbg コマンドを省略すると、マチキャラメーカ V2.0 以降では「背景を表示しない」と解釈します。直前に実行されていたイベントの背景を引き継ぐことはできません。イベント毎に setbg コマンドを設定してください。なお、背景を無効にするための setbg コマンドは設定する必要がなくなります。

### 3. マチキャラメーカ V3.3 以前との差異

以下の項目はマチキャラメーカ V3.3 以前とマチキャラメーカ V4.0 以降では処理方法が異なります。マチキャラコンテンツを作成する際は注意してください。

#### (1)turn コマンド

マチキャラメーカ V4.0 以降では移動イベントを作成すると自動的に turn コマンドが設定されます。しかし、turn コマンドは fix コマンドと同時に設定することはできません。移動イベントに fix コマンドを設定する場合は、自動設定された turn コマンドを削除した後設定してください。

#### (2)バルーン

マチキャラメーカ V3.3 以前では、バルーンに表示する文字列は複数の画像に関連付けることができました。マチキャラメーカ V4.0 以降ではバルーンに表示する文字列は1つの画像にのみ関連付けることができます。同じ文字列を複数の画像で表示する場合は、関連付ける画像の数分の文字列を定義することになります。

### 4. マチキャラメーカでの制限

以下の項目はマチキャラメーカでは制限があります。マチキャラコンテンツを作成する際は注意してください。

#### (1)アクションファイル名

通常アクション、時間反映とも、アクションファイル名に「dummy.tra」を設定することはできません。

## 8. ビヘイビア・モータの概念

マチキャラを作成する際には、これから説明するビヘイビアとモータの概念を理解しておく必要があります。ここではビヘイビアとモータについて具体例を示しながら説明します。

### 8.1. キャラクタの動作

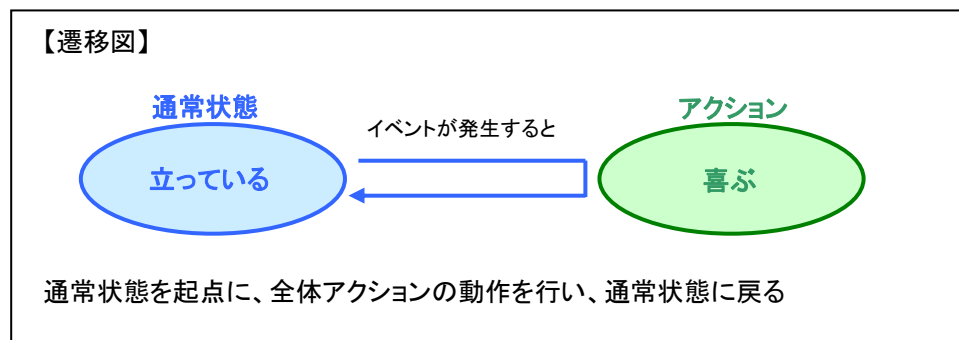
まずはじめにキャラクタをマチキャラとして動作させる場合の動作のさせ方と、その考え方について解説します。

#### 1. キャラクタ全体で動作を行うアクションを考える

キャラクタ全体で動作を行うアクションを全体アクションといいます。

例えば、キャラクタが体全体を使って喜んでいるような動作などは全体アクションと考えられます。キャラクタの動作としては一連の動作を行ってから元の状態に戻るような構成として考えられます。

考え方としてはアクションは状態の遷移として考えられますので、アクションの遷移は通常状態を起点にして、以下のような遷移図で示すことができます。



#### 通常状態 …

マチキャラでは、携帯端末からの要求を受け付けた場合を除き、ランダムにイベントが発生します。本書では、ランダムにイベントが発生している状態を便宜上、通常状態と呼びます。

#### 遷移図 …

想定しているキャラクタのアクション状態の関係を図にしたものです。

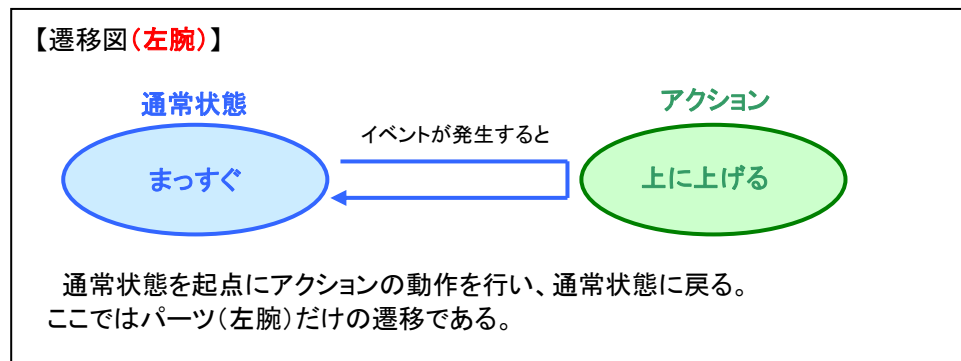
作成しておくキャラクタの状態を整理するのに役立ちます。本書では上図のように記述していますが、書式は自由です。

## 2. キャラクタの一部を動作させるアクションを考える

キャラクタの一部を動作させるアクションをパーツアクションといいます。  
例えば、体はそのまま左腕を上げるなどがパーツアクションになります。

パーツアクションも動作の遷移を通常状態を起点にして、以下のような遷移図で示すことができます。**状態遷移としては左腕だけの遷移のため、パーツアクションを考えるときにはキャラクタのモデルをパーツ毎に分解し、パーツだけの遷移として考えます。**

このアクションを遷移図で表すと以下のように左腕だけの遷移として表せます。



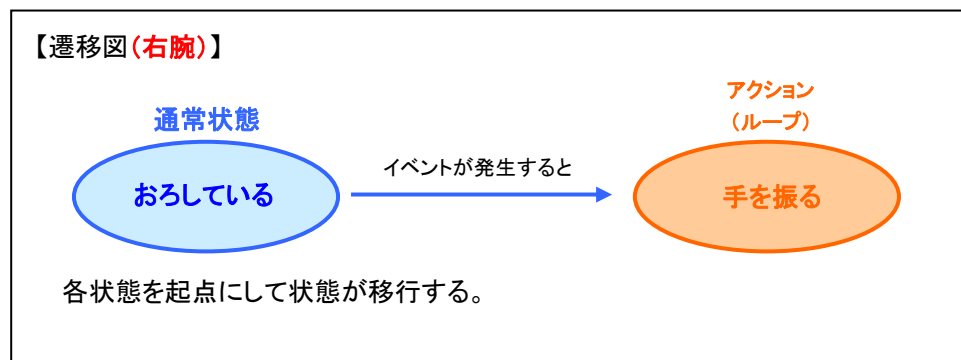
## 3. キャラクタの一部をループ動作させるアクションを考える

同じ動作をきを繰り返し続けるアクションをループアクションといいます。  
例えば、右腕を振り続けるなどがループアクションになります。

ループアクションは動作の開始と終了を同じにすることにより、繰り返しアクションを実行しても不自然にならないようにします。

動作の遷移を通常状態を起点にして、以下のような遷移図で示すことができます。**状態遷移としては右腕だけの遷移のため、キャラクタのモデルをパーツ毎に分解し、パーツだけの遷移として考えます。**

これを遷移図で表すと以下のように右腕だけの遷移として考えられます。  
ここでのポイントは**矢印が一方向である**ということです。



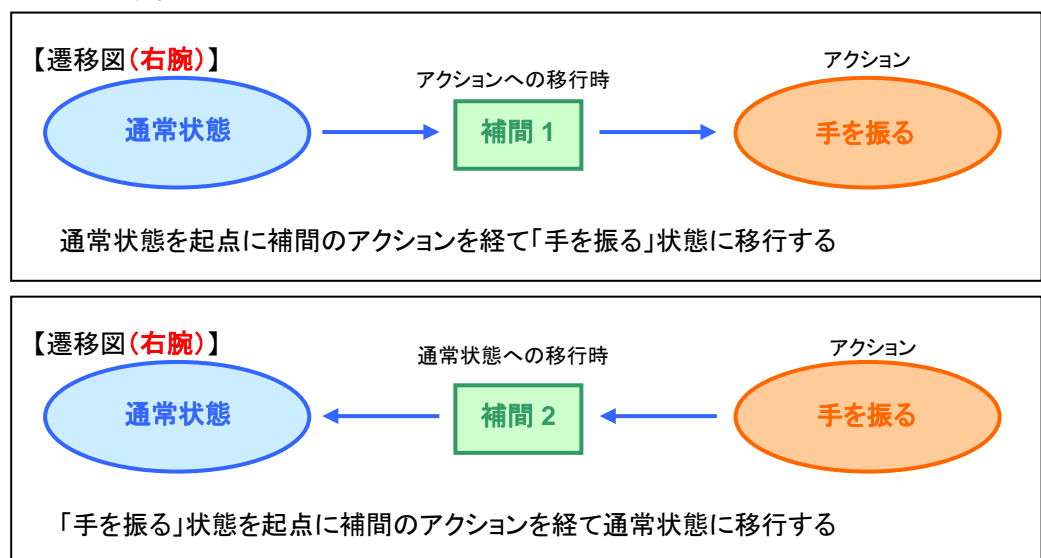
#### 4. 次の動作への「つなぎ」のアクションを考える

アクションとアクションの間が不自然にならないように設定する間のアクションを補間アクションといいます。

例えば、腕を下げている状態と頭の上で腕を振る状態がある場合、下げていた腕が突然頭の上で振られているような動作にならないよう、間に腕を上げる動作を行います。この腕を上げる動作が補間アクションになります。

このように補間アクションを設定することにより、目的とする動作の状態になるまでの間をよりスムーズなアクションの動作を実現することができます。

先ほどの右手を振り続けるアクションをもう一度遷移図で表すと以下のようになります。ここでのポイントは遷移先の状態までの間に補間のアクションを経由していることです。

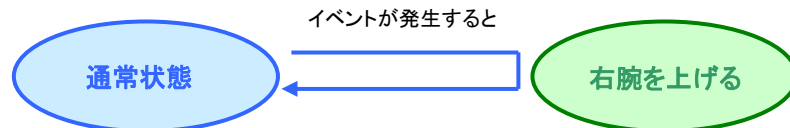


## 8.2. アクションの合成

ここではパーツの動作を同時に行わせることを考えてみます。右腕と左腕を同時に動かす例をみてみましょう。今度はパーツの遷移図の方から考えてみます。

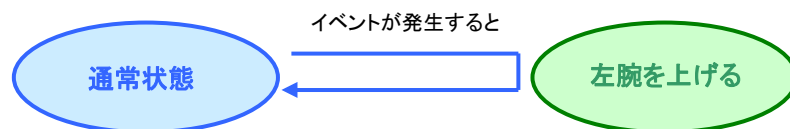
遷移を考えるときはパーツ毎に右腕の遷移と左腕の遷移を分けて考えます。

### 【右腕の遷移図】



通常状態を起点にアクションの動作を行い、通常状態に戻る

### 【左腕の遷移図】



通常状態を起点にアクションの動作を行い、通常状態に戻る

以下のように各パーツ毎のアクションを作成します。

### 【右腕を上げる】



### 【左腕を上げる】

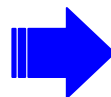


このそれぞれのパーツの動作を同じイベントに割り当てることでこれらを同時に再生することができます。

### 【通常状態】

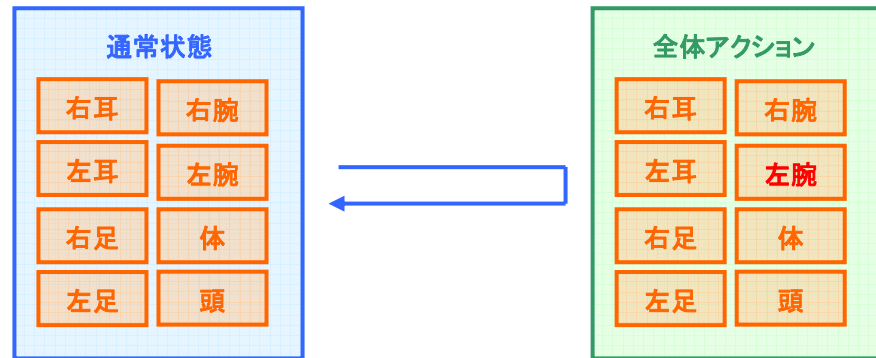


### 【両腕を上げる】



パーツ毎のアクションが同時に動作するので、両腕を上げているアクションになります。このことから、キャラクタ全体を動作させる場合にもパーツ毎の状態の遷移であるとして考えることができます。





キャラクタ全体で動作を行うアクションも各パーツ毎のアクション毎の遷移であるとして考えられます。

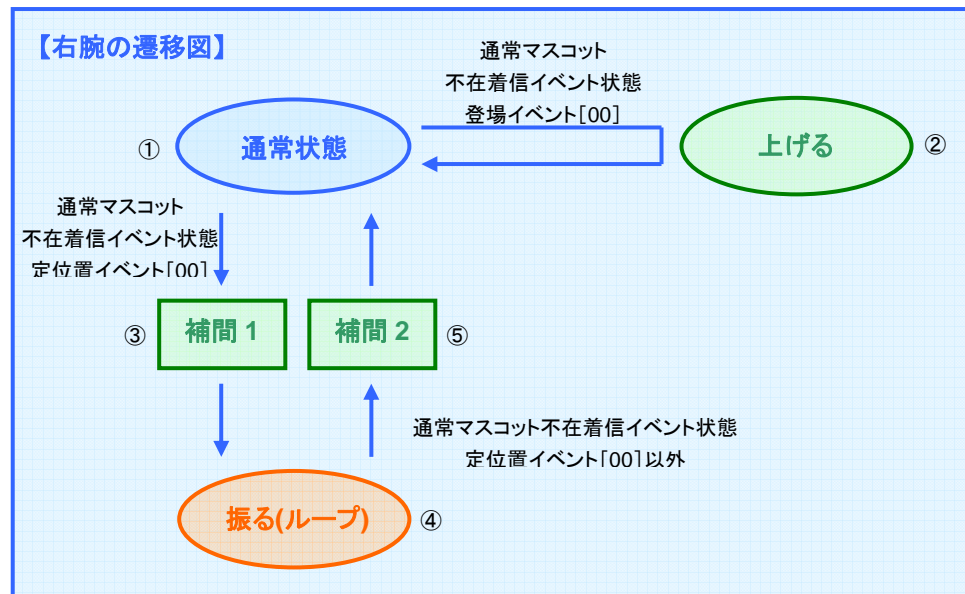
<補足>

この例では、あえて右腕と左腕を別々のパーツとして考えましたが、実際に両腕をあげるだけの動作であれば両腕をひとつのパーツとして考えることも可能です。ここでは考え方を説明するための便宜上のものと捉えてください。

### 8.3. パーツの状態

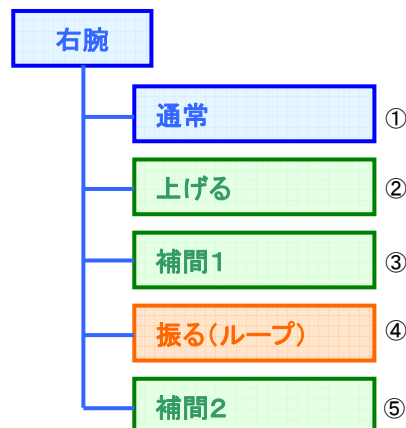
これまで説明してきましたようにアクションを考える際にはキャラクタをパーツ毎に分解して考え、各パーツ毎に動作させてこれらを合成することで一つのアクションを実現できることがわかりました。ここではこれらのパーツの状態について考えてみます。

例としてこれまでの説明に出てきた右腕だけの遷移について考えてみます。右腕の動作としては「右腕を振る(ループ)」と「右腕を上げる」の2つの動作がありました。これらを右腕の遷移図として全体を表現してみます。



上の遷移図は、『右腕は「通常状態」から通常マスコット不在着信イベント状態登場イベント[00]が発行されると、「上げる」状態に遷移し、動作が完了すると「通常状態」に戻ります。また、「通常状態」から通常マスコット不在着信イベント状態定位置イベント[00]が発行されると補間アクションを行って「振る(ループ)」状態に遷移します。「振る(ループ)」状態で不在着信定位置イベント[00]以外のイベントが発行されると補間アクションを行って「通常状態」に戻ります。』と、表現されていることとなります。

この遷移図からわかるように、パーツには「状態」が存在することになります。補間のアクションも状態の一つと考え、①～⑤の5つの状態が存在することになります。



このようにパーツには「状態」が存在していることがわかります。

同様にキャラクタの他の部分もパーツとして考えると左腕、頭、足などの各パーツにも状態が存在することになります。

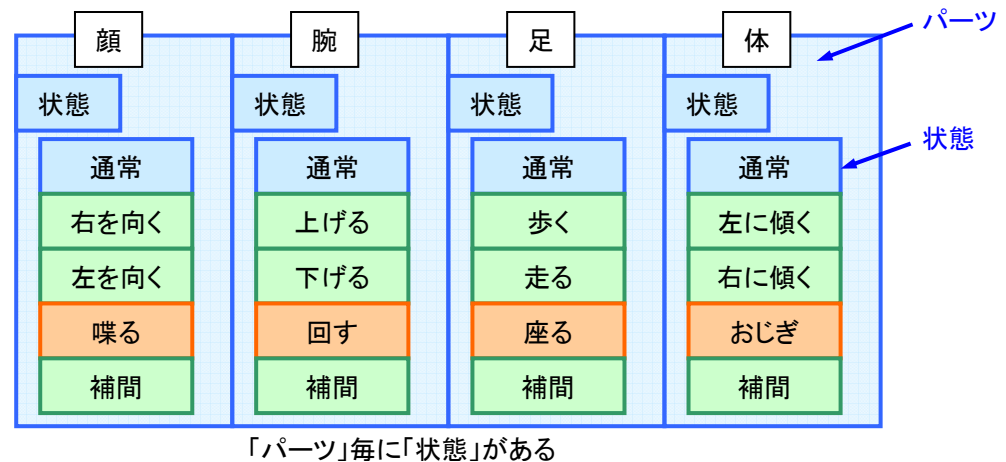
このことから前項の結果に引き続き、キャラクタの動作を考える場合には、

1. アクションの動作を決める
2. 補間のアクションを考える
3. 各アクションで動作させるパーツを決定する
4. 各パーツ毎に遷移図を作成する
5. 遷移図を元に各パーツ毎に状態を洗い出す

という手順でキャラクタの動作を考えて行くということがわかります。

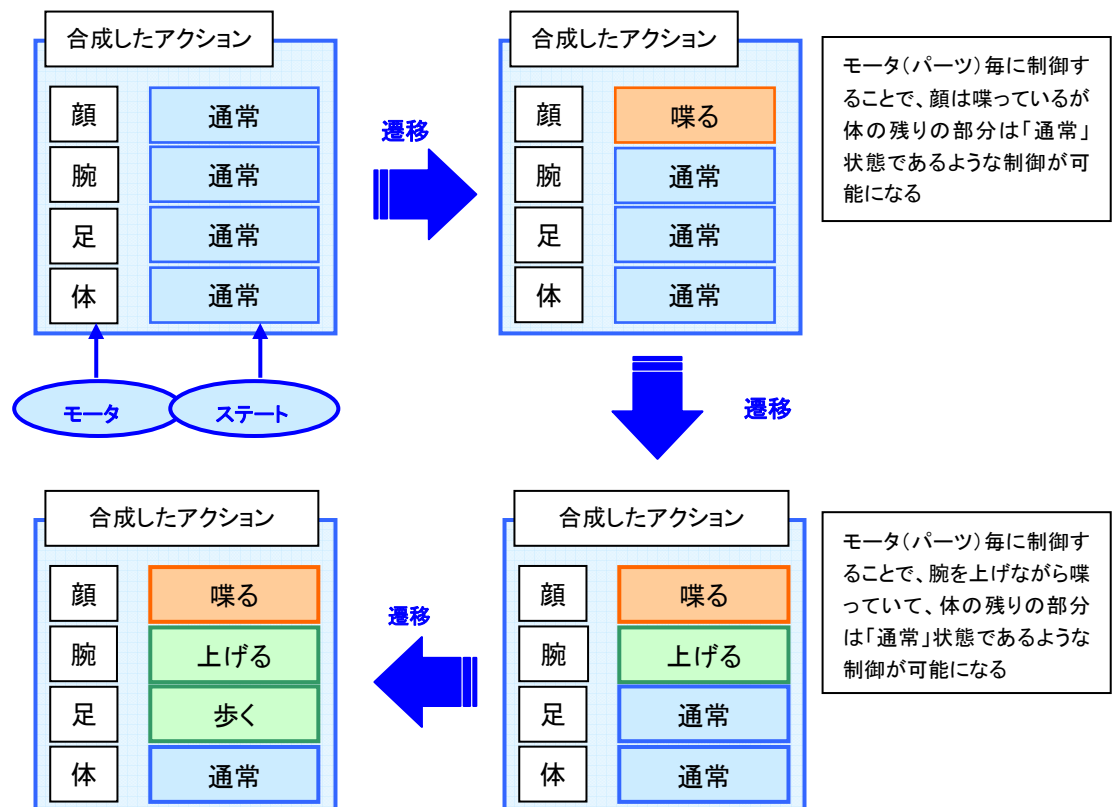
## 8.4. モーターとは

ここではモーターについて説明します。モーターとはこれまでに説明してきた「パーツ」のことをいいます。キャラクタの動作を「各パーツ毎のアクションの合成」としてとらえ、各パーツ毎の「状態」の遷移によりひとつの動作を実現することになります。



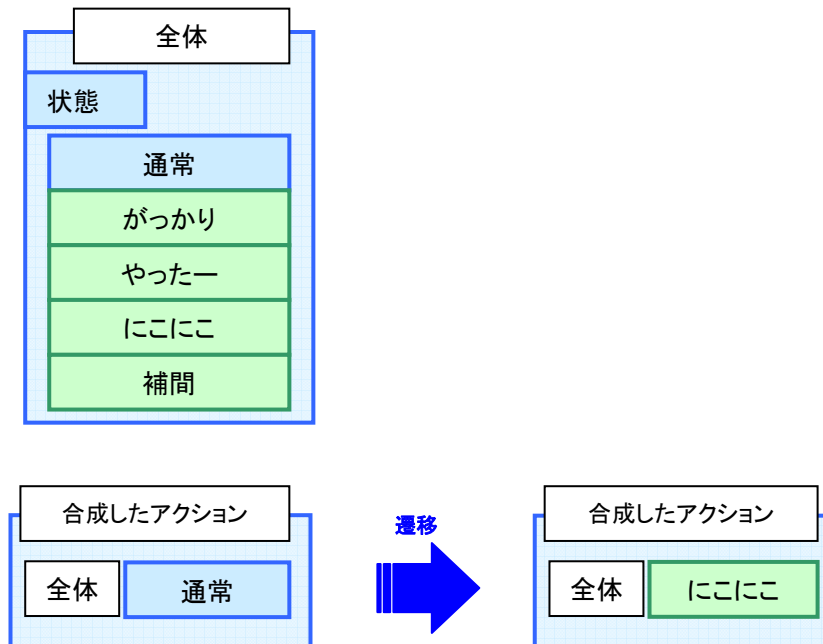
上記のように各パーツ毎に状態を用意した場合、この各パーツのことを**モーター**と呼びます。また、各モーターに用意した状態のことを**ステート**と呼びます。

実際にアクションを行う際は各**モーター**(パーツ)毎の**ステート**(状態)を合成することでアクションを実現します。



モーター(パーツ)毎のステート(状態)を合成することで、さまざまな動作が実現できます

モータは必ずしも複数用意する必要はなく、一つのモータで動作を実現しても構いません。これはキャラクタをパーツに分けずに、一度に全てのパーツの状態を遷移させて、一つ一つのアクションとして作成することになります。



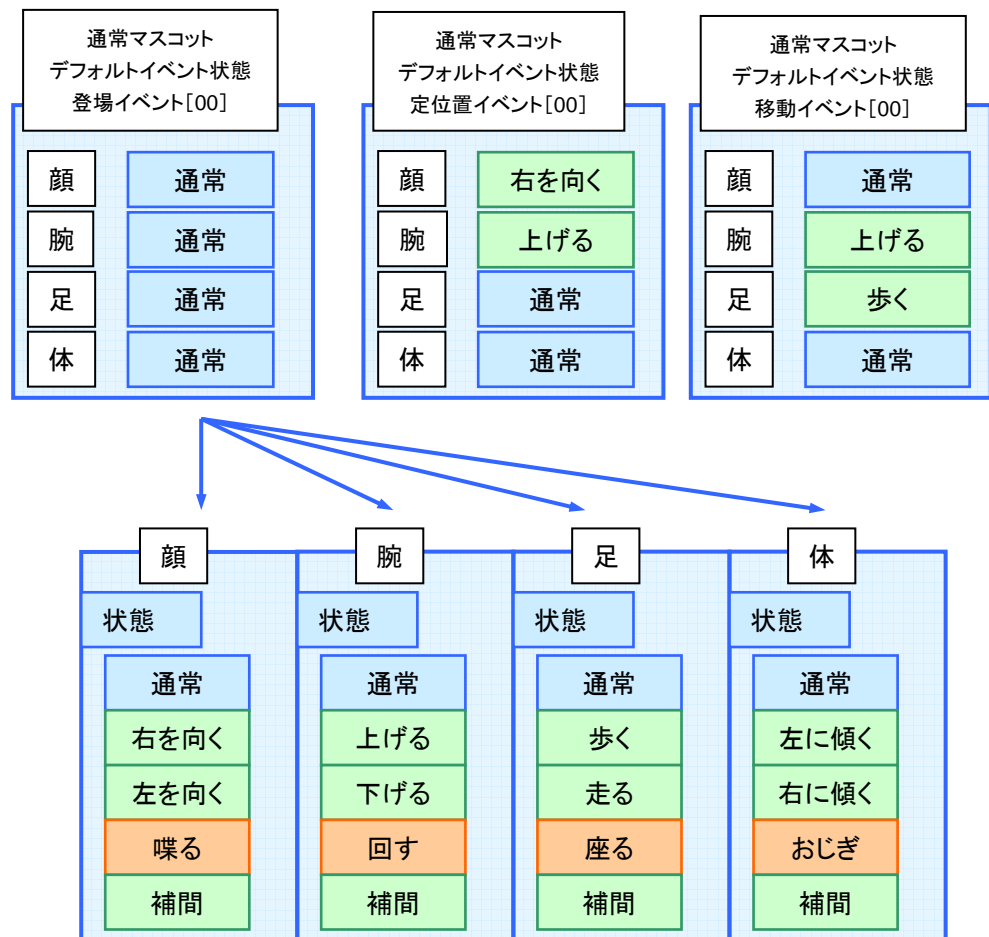
モータを一つとして考える場合はキャラクタの表現を一つのステート(状態)で用意することになります。著作権のある動き方の決まったキャラクタを動作させる場合にはモータを一つにした方がそのキャラクタ本来の動きに忠実になる場合もあります。

マチキャラメーカーでは、モータ(パーツ)に対するステート(状態)を定義して行くことでマチキャラの動作を作り上げて行くことになります。

✓ 2D マチキャラは、複数のモータは設定できません。

## 8.5. ビヘイビアとは

続けてビヘイビアについて説明します。ビヘイビアとはイベントに応じたキャラクタの振る舞いのことを指します。イベントとは携帯端末からの要求や、時刻など、キャラクタの状態を遷移させるためのトリガーになるものをいいます。マチキャラでは、キャラクタの各要求を受け付けた時をイベントとして捉えてこれをビヘイビアから設定します。具体的には、今までの説明に出てきた各モータのステートをそれぞれのイベントに応じて設定することになります。



ビヘイビアでは「通常状態」等のマチキャラの状態に応じて各モータ（パーツ）のステート（状態）を定義することにより、マチキャラ全体の制御が可能になります。

### イベント …

携帯端末上で各イベントが発生した時の状態で、この時の各パーツの状態を定義します。実際にはこの「イベント」の各イベントコードに対応した動作を設定することでマチキャラにさまざまな表現をさせることになります。マチキャラが携帯端末上にロードされた直後は通常マスコットデフォルトイベント状態登場イベントが実行されます。

## 8.6. ビヘイビアとモータの詳細

ここではビヘイビアとモータについてもう少し詳しく見ることにします。

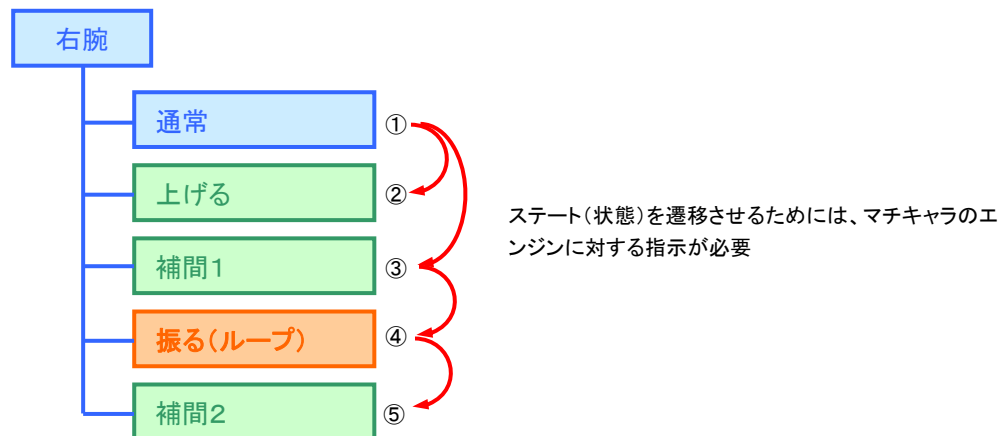
ビヘイビアがマチキャラ全体の制御を行っており、イベントの種類や、モータのステートの遷移の方法、ステートで指定できるイベントやコマンドについての説明になります。

キャラクタの動作を考える時には、キャラクタをパーツに分けて考え、それぞれのパーツの動作の組み合わせによって各それぞれの動作をさせるように考えるか、キャラクタ全体をひとつのパーツとして動作をさせるように考えるかの2種類の考え方がります。いずれにしてもモータにステート(状態)を用意し、その状態を遷移させることでキャラクタの動作が行われることになります。ここでは、モータ(パーツ)におけるステート(状態)の遷移について説明します。

### 1. イベントに応じたステート(状態)の遷移

マチキャラの内部ではマチキャラの動作を制御しているマチキャラのエンジンがあります。このマチキャラのエンジン内部では、マチキャラを動作させるためにモータ(パーツ)毎の現在のステート(状態)を保持しています。従って、実際にステートを遷移させるためには、マチキャラのエンジンに対してステートの遷移を指示することによりステートの遷移を行わせることになります。

＜モータ(右腕)での例＞



ステートの状態を遷移させるにあたり、「いつ」「どのように」ステートを遷移させるかを指定する必要があります。マチキャラではこの遷移のタイミングをビヘイビアとモータのそれぞれのタイミングで制御することになります。ここではビヘイビアから指定するステートの遷移のタイミングを説明します。ビヘイビアから指定するステートの遷移タイミングとしては以下のものがあります。

#### 1. イベント発生時

イベントには大きく分けると 2 種類あります。「外的要因イベント」と「内的要因イベント」です。携帯端末からの要求で発行されるイベントを「外的要因イベント」と呼びます。エンジンがランダムに自動発行するイベントを「内的要因イベント」と呼びます。

＜外的要因イベント＞

##### (1) アラームイベント

スケジューラのアラーム鳴動時に発生するイベントです。

ポップアップイベントです。

##### (2) ダイアログイベント

音声対話機能を利用中に発生するイベントです。

ポップアップイベントです。

(3)i コンシェル TOP ショートカットフォーカスイベント

待受画面上の i コンシェルアイコンにフォーカスが当たった時に発生するイベントです。

ポップアップイベントです。

(4)i コンシェルイベント

移動機の i コンシェルサービスに対応して発生するイベントです。

ポップアップイベントです。

(5)i コンシェル TOP イベント

i コンシェル TOP の画面上で発生するイベントです。

(6)不在着信通知／履歴確認イベント

音声またはテレビ電話の不在着信があった場合に発生するイベントです。

(7)未読メールありイベント

未読メール数が指定件数に達した場合に発生するイベントです。

(8)メール受信通知／履歴確認イベント

メールの受信があった場合に発生するイベントです。

(9)通話時間イベント

通話時間が指定時間に達した場合に発生するイベントです。

(10)隠しマスコットイベント

隠された外見に変更するイベントです。携帯端末によって発生条件は異なります。

隠しマスコットイベントに対応するかは携帯端末メーカーの選択により決定します。

(11)メーカーイベント

携帯端末メーカー固有のイベントです。携帯端末のメーカーが独自に取り決めたトリガーにより発生します。イベント ID は携帯端末メーカーが取り決めます。

メーカーイベントに対応するかは携帯端末メーカーの選択により決定します。

(12)デフォルト

起動時、リセット発生時などいずれのイベントも発生していない状態です。

### <内的要因イベント>

(1)登場イベント

マスコットデータロード時およびマスコットの場面遷移時に発生するイベントです。

このイベント実行後は定位置イベント、移動イベント、癖イベントがランダムに発生されます。

(2)移動イベント

マスコットが携帯端末上の位置を移動するイベントです。内部イベントです。

移動ポイントがランダムに決定される自動移動と、指定された移動データに従って移動する定型移動があります。自動移動にするか定型移動にするかはコンテンツ作成者が決定します。

(3)定位置イベント

携帯端末上の定位置で自動発生する内部イベントです。

定位置イベントの実行時間はランダムに決定されます。

(4)癖イベント

エンジン側で常に10%の確率で自動発生する内部イベントです。

### <その他のイベント>

(1)時間反映イベント

システム日付時間の変化とコンテンツの時間反映条件を使用してマスコットの表示を一時的に一部変更するイベントです。キャラクタの動作はこのイベントによる影響を受けません。

(2)時間指定イベント

携帯端末に登録されているスケジュールの日時および所有者情報の誕生日になった場合にマスコットの表示を一時的に一部変更するイベントです。それぞれのスケジュール属性に対して設定することができます。

それぞれのイベントが発行されると、そのイベント ID に応じてキャラクタを動作させる記



ですが、それぞれのアクションによって割当てイベント ID にはルールがあります。

【定義可能なイベント ID】

イベント種類	イベントコード			
	ブロック1	ブロック2 (イベント状態)	ブロック3	ブロック4
デフォルト	D(通常マスコット) Sxx(時間指定 xx マスコット) xx: スケジュール属性(00~99)	D	S(登場) M(移動) F(定位置) H(癖)	00~99
アラーム		W** (**はアクション種別)	S(登場) M(移動) F(定位置) H(癖)	00~99
ダイアログ		X** (**はアクション種別)	S(登場) M(移動) F(定位置) H(癖)	00~99
i コンシェル TOP ショートカット フォーカス		Y** (**はアクション種別)	S(登場) M(移動) F(定位置) H(癖)	00~99
i コンシェル		A** (**はアクション種別)	S(登場) M(移動) F(定位置) H(癖)	00~99
i コンシェル TOP		Z** (**はアクション種別)	S(登場) M(移動) F(定位置) H(癖)	00~99
不在着信通知／履歴確認		P	S(登場) M(移動) F(定位置) H(癖)	00~99
未読メールあり通知／履歴確認		N	S(登場) M(移動) F(定位置) H(癖)	00~99
メール受信通知／履歴確認		M	S(登場) M(移動) F(定位置) H(癖)	00~99
通話時間通知／確認		C	S(登場) M(移動) F(定位置) H(癖)	00~99
メーカー		0~9, #からなる 8 文字 + メーカー識別番号 2 文字	S(登場) M(移動) F(定位置) H(癖)	00~99

隠しマスコット		0～9, #からなる最大 8 文字	S(登場) M(移動) F(定位置) H(癖)	00～99
---------	--	-------------------	----------------------------------	-------

イベントの定義を行う際には上記のイベント ID を意識する必要はありません。

一般的にパーツを複数用意した場合(モータを複数用意した場合)には全体のアクションでは各パーツの状態を定義することでアクションするように作成します。また、パーツが一つだけの場合(モータが1つの場合)は一つの状態でアクションを行うように作成し、その状態を定義します。

注: メーカーイベントおよび隠しマスコットイベントは各携帯端末メーカーによりイベントIDおよびイベント発行タイミングを取り決めます。

## 2. イベント解除時

マチキャラでは以下の場合に現在のイベントが解除されます。

- (1) i コンシェルイベント状態の解除
  - i コンシェルイベント状態のときに i コンシェル TOP 画面に遷移するとデフォルトイベント状態に遷移します。
- ✓ 待ち受け画面で i コンシェルイベントを実行中にポップアップを選択すると i コンシェル TOP 画面に遷移します。
- (2) 履歴確認操作によるイベント状態解除
  - 不在着信イベント状態のときに不在着信履歴確認を発行するとデフォルトイベント状態に遷移します。未読メールありイベント状態、メール受信イベント状態のときにメール受信履歴確認を発行するとデフォルトイベント状態に遷移します。
- (3) 通話時間イベント状態の解除
  - 通話時間イベント状態に遷移後5分経過した場合、通話時間イベント状態は解除され、デフォルトイベント状態に遷移します。
  - 通話時間イベント状態のときに通話時間イベントが発生した場合、現在のイベント状態は一度解除されます。
  - 新しい通話時間イベントと条件が適合するイベントが存在する場合は、新たに通話時間イベントが実行されます。この時点から5分が新しいイベントの有効時間となります。
  - 新しい通話時間イベントと条件が適合するイベントが存在しない場合は、デフォルトイベントが実行されます。
- (4) リセット操作によるイベント状態解除
  - どのイベント状態にあってもリセットイベントを発行すると通常マスコットデフォルトイベント状態に遷移します。時間指定イベント、メーカーイベントおよび隠しマスコットイベントもイベント状態が解除されます。
  - この機能の実装／未実装および実装方法は携帯端末メーカーが取り決めます。
- ✓ メーカーイベントおよび隠しマスコットイベントは携帯端末メーカーにより開始／解除トリガーを取り決めます。前述以外のタイミングでイベント状態が解除される場合があります。
- ✓ イベント優先順位が高いイベントが発生した場合にも現在のイベント状態から他のイベント状態へ遷移します。
- ✓ マチキャラ V4.0 以降では、ポップアップイベントを実行中に特定の画面に遷移した場合に強制的に実行中のイベントを解除します。

遷移先画面 イベント	待ち受け画面	メニュー画面	ブラウザ画面	＜電話画面	メーカー画面	メール受信確認画面	i コンシェル TOP 画面	音声入力モード画面	その他画面
アラームイベント								×	
ダイアログイベント	×	×	×		×		×		×
i コンシェル TOP ショートカットフォーカスイベント		×	×		×		×	×	×
i コンシェルイベント								×	

i コンシェル TOP イベント	×	×	×		×			×	×
------------------	---	---	---	--	---	--	--	---	---

### 3. イベントの優先順位

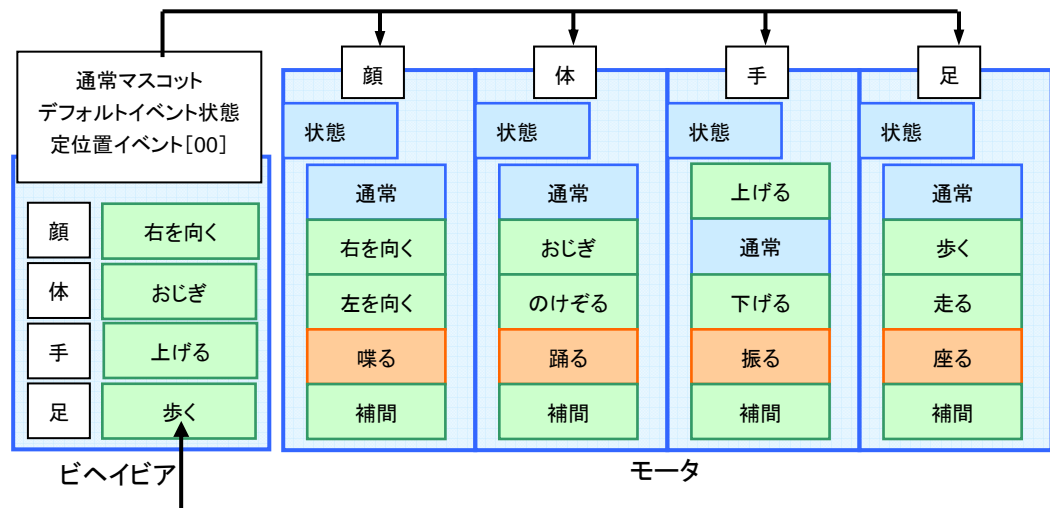
イベントの優先順位を以下に示します。

- (1) アラームイベント、ダイアログイベント、i コンシェル TOP ショートカットフォーカスイベント
- (2) i コンシェルイベント、i コンシェル TOP イベント
- (3) 不在着信イベント
- (4) 未読メールありイベント
- (5) メール受信イベント
- (6) 通話時間イベント
- (7) メーカーイベントイベント(携帯端末メーカーで設定)

隠しマスコットイベントおよびデフォルトイベントはイベント優先順位判定は行いません。

### ＜モータが複数ある場合＞

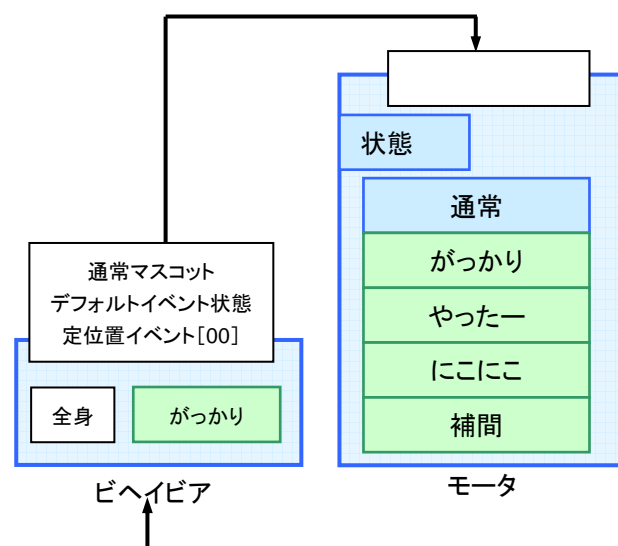
モータが複数ある場合は各モータにそのパーツ毎の状態を用意し、ビヘイビアから各モータ毎の状態を定義することによりアクションを実現します。



モータに用意した各状態をビヘイビアから定義することにより「通常マスコットデフォルトイベント状態定位置イベント[00]」が発行された時の状態を設定することができる。

### ＜モータが1つの場合＞

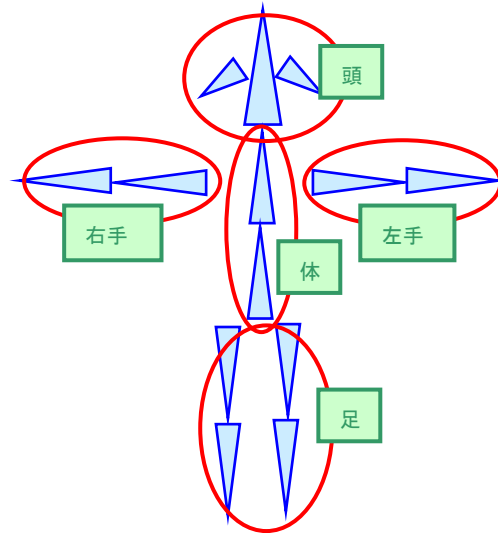
モータが1つの場合は1つのパーツに複数の状態を用意し、ビヘイビアからはその特定の状態を指定することでアクションを設定します。



モータがひとつの場合はモータに用意した特定の状態をビヘイビアから定義することにより「通常マスコットデフォルトイベント状態定位置イベント[00]」が発行された時の状態を設定することになる。

上記のように各パーツ毎に動作を定義する場合にはアクションを作成する際に以下のような点に注意する必要があります。

例えば、以下のように複数のパーツ(モータ)を使用して各パーツを管理する際には重複したキーフレームのあるボーンを作成しないようにする必要があります。



各パーツ毎に動かすボーンを決めて頭のアクション作成時には、頭として決めた部分以外のボーンにキーフレームを作成しないようにする必要があります。

#### キーフレーム …

キャラクターの位置等の情報を設定したフレーム。3D アニメーション作成ツールでは、いくつかのキーフレームを設定することにより、キーフレーム間の動作を補間して自動的にアニメーションを作成します。

## 2. ステート(状態)遷移の種類とその方法

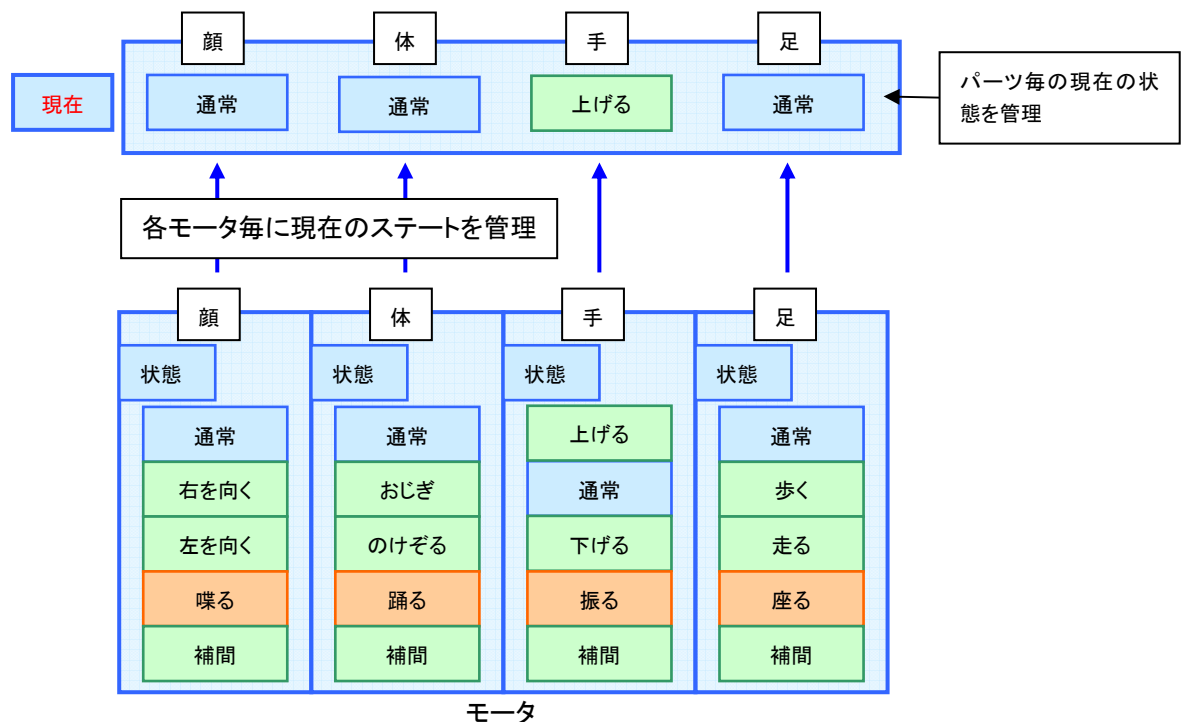
ここではさらにビヘイビアからの各モータの状態を変更する方法の種類の違いについて説明します。

マチキャラの内部ではマチキャラの動作を制御しているマチキャラのエンジンがあります。このマチキャラのエンジン内部ではキャラクタの各モータについて、「**現在状態(現在のステート)**」と「**目的状態(遷移先のステート)**」という2つの状態を管理しています。マチキャラではこの2つの状態を変更することによりキャラクタのパーツ毎のステートの遷移を実現しています。実際にマチキャラのデータを作成する際にはこの概念についてもある程度の理解が必要になってきます。ここではまずこの2つの状態の管理方法について説明します。

### (1) マチキャラが管理する「**現在状態(現在のステート)**」

マチキャラの内部では、キャラクタのパーツ(モータ)が今どのステートなのかを各パーツ毎に管理しています。

マチキャラ内部の「**現在状態(現在のステート)**」

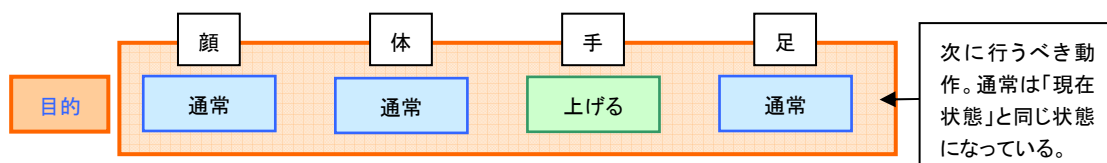


マチキャラが内部で管理している「**現在状態**」とは各パーツ毎の現在のステートのことで、現在画面に表示されている「現在のパーツ毎の動作」のことになります。上記の例では現在のステートに設定されている動作、すなわち「キャラクタの顔・体・足は通常の状態、手を上げている状態」が画面に表示されており、「現在のステート」であることとなります。

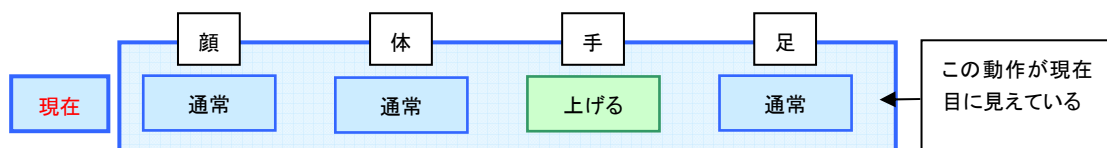
## (2) マチキャラが管理する「目的状態(遷移先のステート)」

マチキャラでは、「現在状態」とは別に「次にどの動作をすべきか」を管理するために、マチキャラの内部に「目的状態(遷移先のステート)」というものパーツ毎に管理しています。これはキャラクタの各パーツが次にどの状態に遷移すべきかを管理しており、補間のアクションのように複数の動作を経てある状態に達する動作を行いたい場合に設定を行うものです。「目的状態」が変化するとマチキャラは「現在状態」から「目的状態」に遷移しようとします。これにより現在の状態からの動作が開始されることになります。通常、何も設定されていない場合は「現在状態」と「目的状態」は同じ状態になっています。

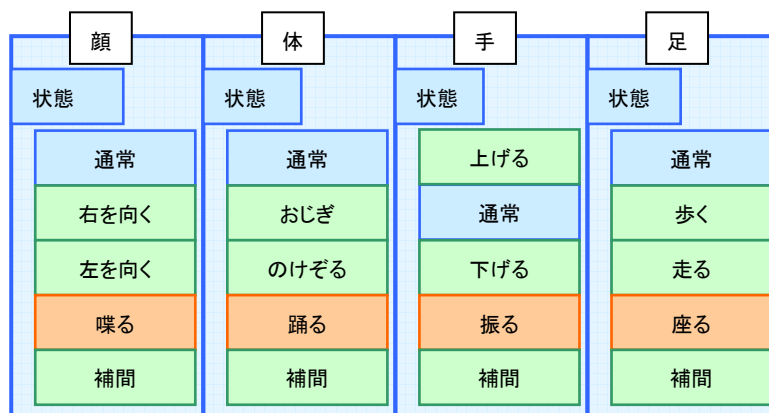
### マチキャラ内部の「目的状態(遷移先のステート)」



### マチキャラ内部の「現在状態(現在のステート)」

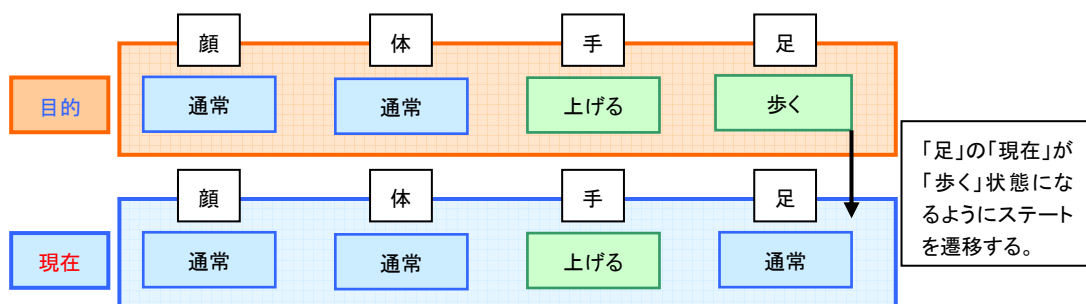


各モータ毎に現在のステートを管理



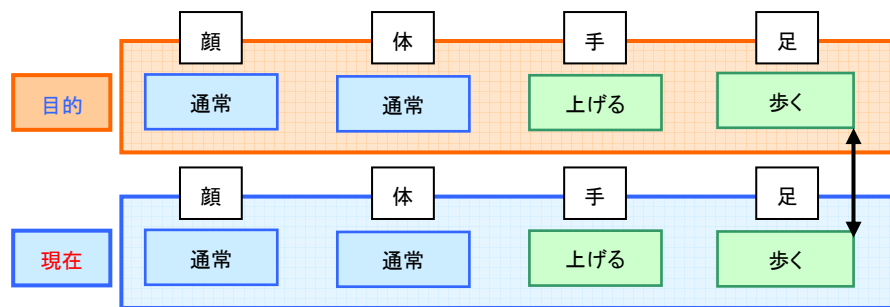
モータ

マチキャラは「目的状態」が変更されると、「現在状態」と「目的状態」が同じになるように状態の遷移を開始します。(状態を変更し始めます)



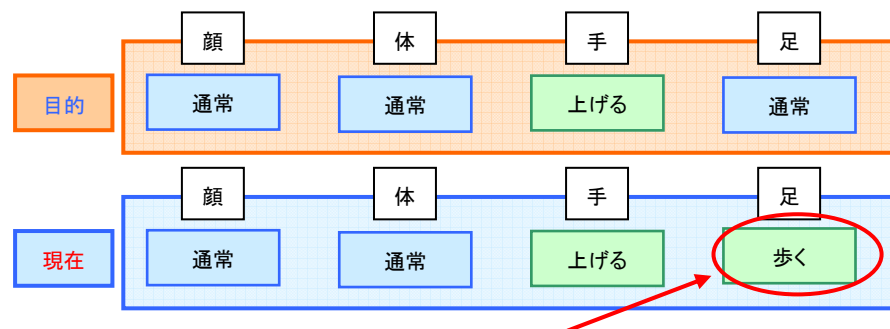
このように「目的状態」に遷移先のステートを設定することで画面に見えているキャラクタの動作が開始されることになります。上図の例では動作を開始して、「足」が「歩く」状態になるように状態の遷移をし続けます。





「足」の「現在状態」が「歩く」状態になった時点でステートの遷移は終了します（厳密には、「歩く」状態への遷移を繰り返します）。この時画面に見えているキャラクタは「顔・体は通常の状態、手は上げており、足は歩いている」動作をしていることになります。

「目的状態」に次の状態を設定するとキャラクタの状態を遷移させることが分かりましたが、**「現在状態」を変更することによっても状態を遷移させることができます。**



「現在状態」を変更した場合には、キャラクタは指定されたステート（状態）にすぐに遷移します。画面に見えているキャラクタは指定された動作を即時に開始します。「目的状態」を変更した場合との違いは、以前に出てきた**「目的の動作を行うまでの補間のアクションの動作を伴わない」**という点です。

上記までの実際の例では以下のような違いが出てきます。

- a. **「現在状態」**の「足パーツ」を「歩く」状態に変更した場合  
画面に見えているキャラクタの足は最初は通常状態だったものが突然歩く動作を行っている状態になります。（急に歩きだす）
- b. **「目的状態」**の「足パーツ」を「歩く」状態に変更した場合  
画面に見えているキャラクタの足は最初は通常状態だったものが、まず、補間のアクションの動作（この例であれば歩きだすような動作）を行ってから歩いている状態になります。（スムーズに歩きだす）  
※補間のアクションはモータで遷移先を設定することにより実現します。（後述）

ここまでの説明でキャラクタの各パーツの状態を遷移させるには、「現在状態」を変更する方法と、「目的状態」を設定する方法の2種類があることが分かりました。

マチキャラメカでは、キャラクタのパーツの状態を遷移させる方法をそれぞれ以下のように呼び、区別しています。

状態の変更方法	意 味
「実行」(set)	「 <b>現在状態</b> 」の状態を変更します。
「設定」(play)	「 <b>目的状態</b> 」の状態を変更します。

### 3. モータにおけるステート(状態)遷移

今までの説明ではビヘイビアでは「現在状態」や「目的状態」に各モータのステート(状態)を変更することで状態が遷移するということについて説明してきましたが、今度はその状態の変化に応じたステートの遷移について説明します。

ステートでは以下のように各ステートのタイミングに応じたイベントを設定することができるようになっています。

#### <ステートで設定できるイベント>

イベント種類	意 味
開始(start)	「現在状態」が指定されたステートに到達した時
終了(end)	「現在状態」のアクションが終了した時
イベント(event)	「目的状態」が変更された時
間隔(time)	「現在状態」がステートに到達してからの経過時間を迎えた時

各イベントでは状態の遷移のタイミングを取得することができますので、そのタイミングで何をするかの指定を行う必要があり、これを「コマンド」と呼びます。各イベントで指定可能なコマンドは以下のとおりです。

#### <ステートのイベントで指定できるコマンド>

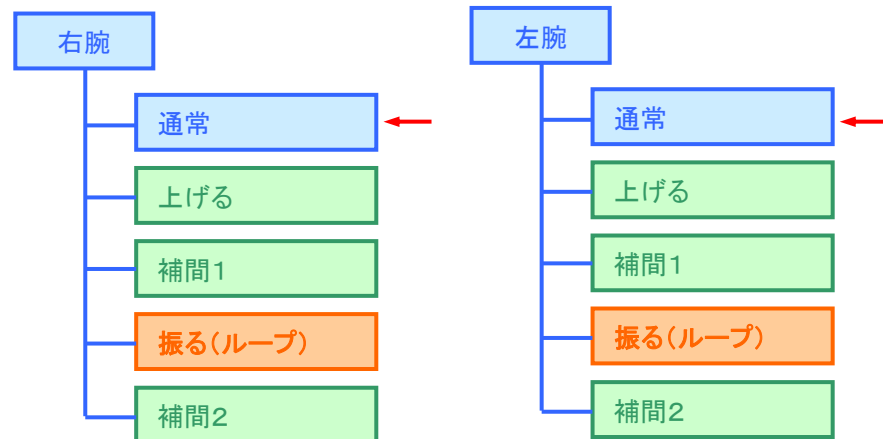
イベント種類	意 味
前景(setfg)	前景画像を設定する(透過が行われます)
背景(setbg)	背景画像を設定する(透過は行われません)
遷移(goto)	指定されたステートに遷移する
遷移先設定(next)	遷移先のステートを設定する(実際の遷移は行わない)
遷移先へ(gotonext)	保存されている遷移先のステートに遷移する(設定されている遷移先がない場合は現在のステートに戻る)

#### ・ステートの遷移方法

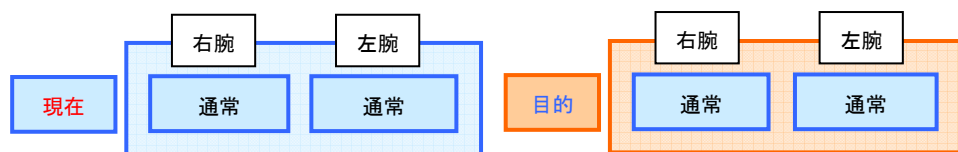
各ステートではイベントの指定を行うことで、そのステートの中で「いつ」「何をするか」ということが指定できます。また、ビヘイビアでの「現在状態」や「目的状態」の変更によってもステートの遷移が行われますので、このあたりも合わせて説明します。以下に例を挙げて説明します。

以下の例では、右腕と左腕の両モータについて前項で説明したマチキャラ内部の管理方法と併せてどのように遷移するのかを説明します。

＜右腕と左腕モータのステートの例＞



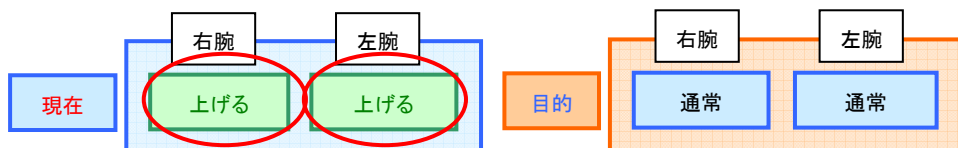
上の図では、右腕と左腕のモータに存在するステートを記述してあります。また、下の図では前項で説明したパーツ(モータ)毎の「現在状態」と「目的状態」がそれぞれ「通常」である状態を表しています。



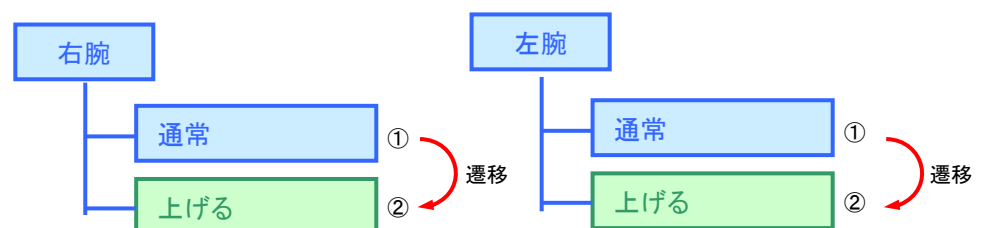
この状態ではマチキャラの画面にはそれぞれ右腕・左腕ともに「通常」に設定されているアクションが実行されて表示されていることになります。

注: 実際には右腕と左腕だけのモータということはありませんが説明のため便宜上2つのモータで説明します。

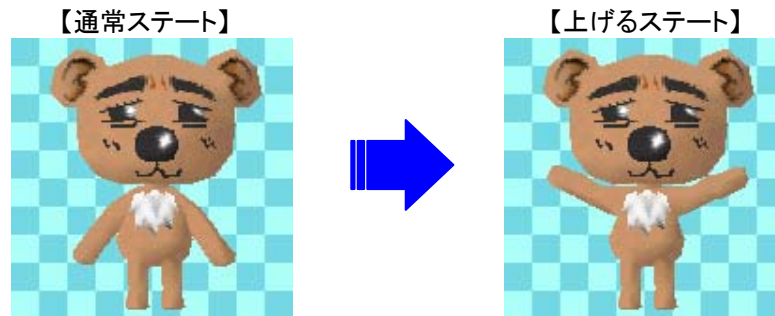
ここで、携帯端末からのイベントとして通常マスコットデフォルトイベント状態定位イベント[00]が発行された場合を考えます。ビヘイビアでは携帯端末からのイベントを取得してマチキャラ内部の「現在状態」や「目的状態」を変更しますので、ここでは「現在状態」の「右腕」「左腕」のそれぞれの状態を「上げる」に変更したとします。



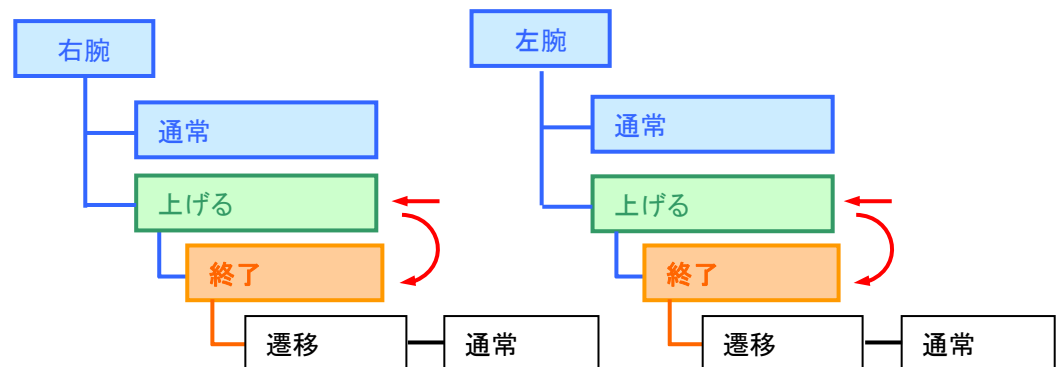
ビヘイビアで「実行(set)」によって「現在状態」の「右腕」「左腕」を「上げる」に変更します。すると、「現在状態」が変更されたので、ステートの遷移が行われます。



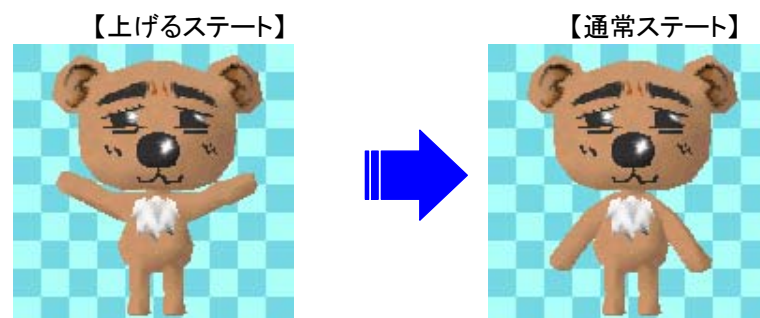
「現在状態」の変更によって各モータの遷移が行われます。この例では両腕のモータがそれぞれ「上げる」状態に遷移することになります。これによりマチキャラの画面にはキャラクタが両腕をあげている動作が表示されます。



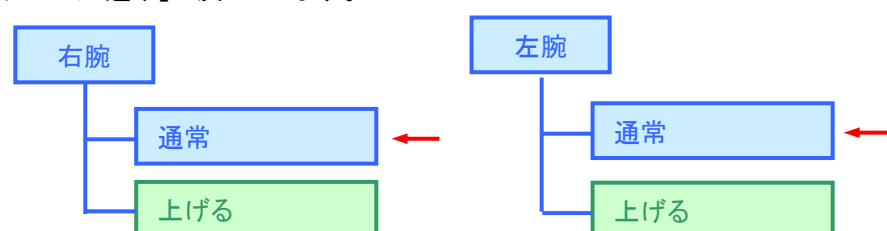
次に、キャラクタが両腕を上げ終わった後にまた通常の状態に戻すことにします。その場合にはステートにアクションの終了時のイベントを設定することで状態の遷移を行います。

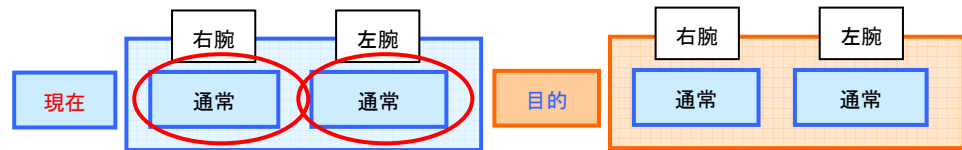


ステート「上げる」にイベントとして「終了」を設定し、コマンド「遷移」の遷移先として「通常」を指定します。このことにより「上げる」ステートにおけるアクションの終了時に「通常」に遷移することになります。



ステートでの終了イベントに「通常」ステートへの遷移の指定があるため、各モータのステートは「通常」に戻っています。



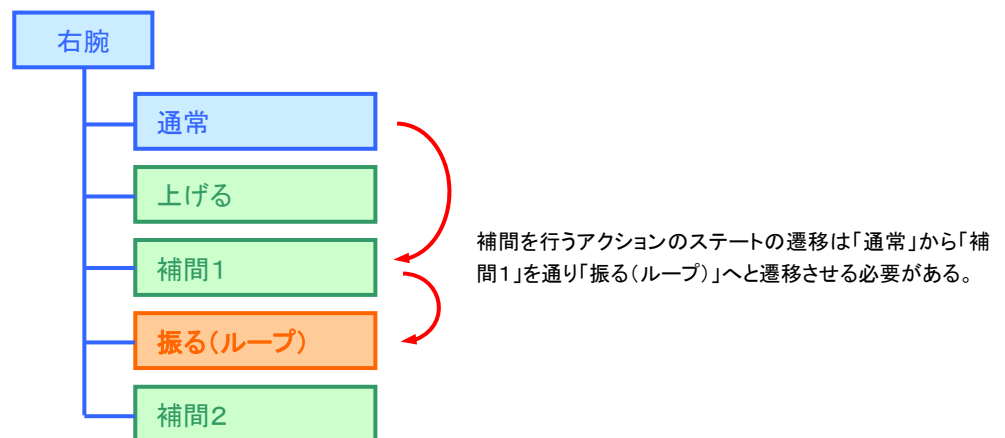


次に、ループするアクションの動作を見てみます。

携帯端末から「右腕だけを振る」というアクションのイベントが発行されたとします。イベントを通常マスコットデフォルトイベント状態移動イベント[00]とします。

ループするアクションでは補間のアクションを行う必要がありますので、ステート側での遷移は以下のように遷移させる必要があります。

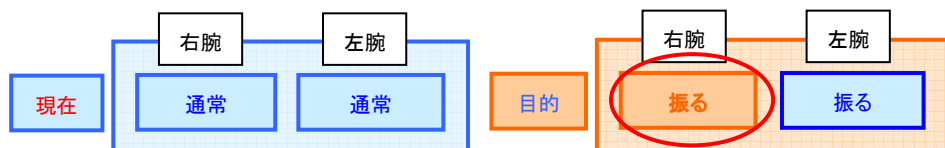
#### <右腕のモータ>



「通常」ステートから「補間1」アクションを行い「振る(ループ)」アクションを行うことでスムーズなアクションへの移行となります。

補間アクションが存在する場合はビヘイビアではイベントを取得した時に「目的状態」を設定するようにしておきます。

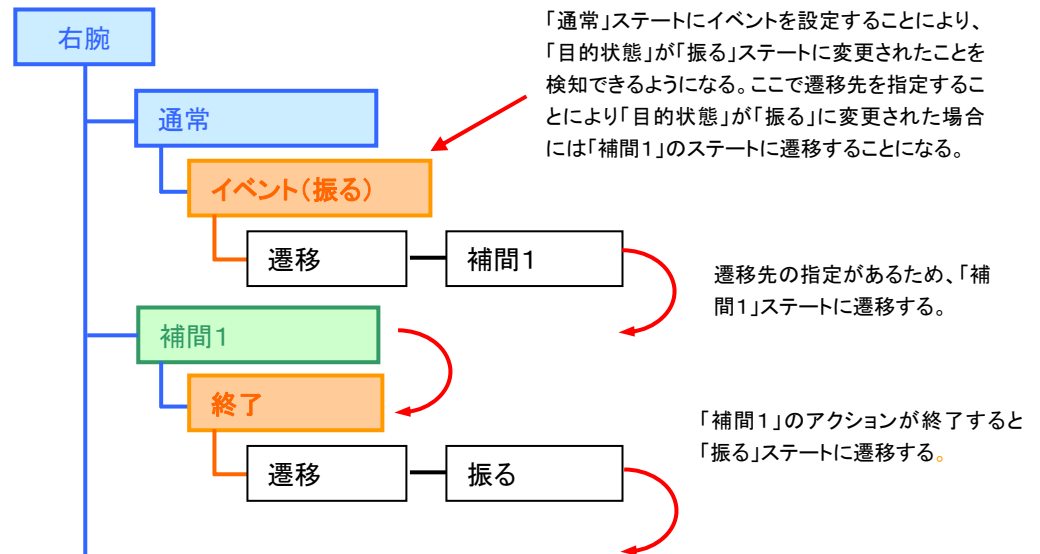
#### <現在状態と目的状態>



ビヘイビアで「設定(play)」によって「目的状態」の「右腕」を「振る」に変更します。すると、変更された「右腕」について、ステートの遷移が行われます。この時、変更された内容が「目的状態」であるため、ステートの遷移は**そのモータに設定されている内容に従って遷移を行います**。

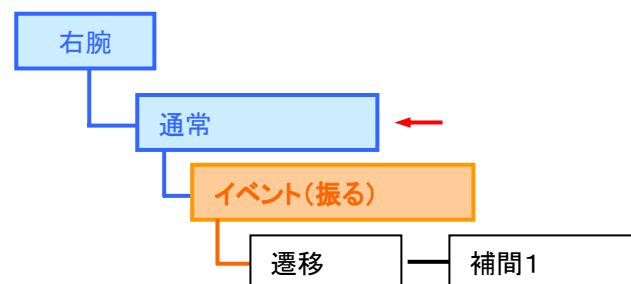
そのため、ステートでは目的状態が変更された時のイベントを取得してステートを遷移させます。

### <右腕モータのステートの例>



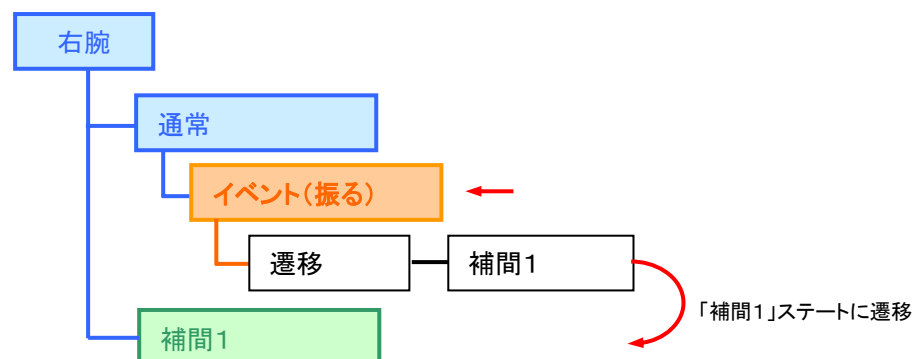
この場合、「通常」ステートには目的状態の変更が行われたかどうかを検知するイベントを設定しておきます。イベントとして「振る」への目的状態の変更が行われたかどうかを検知するイベントを設定しておく、ビヘイビアで目的状態が変更された場合に変更されたことを検知してステートを遷移させることができます。

### <「通常」状態>



ビヘイビアより「目的状態」が「振る」に変更されると、「通常」ステートでこのイベントをハンドリングして「補間1」に遷移します。

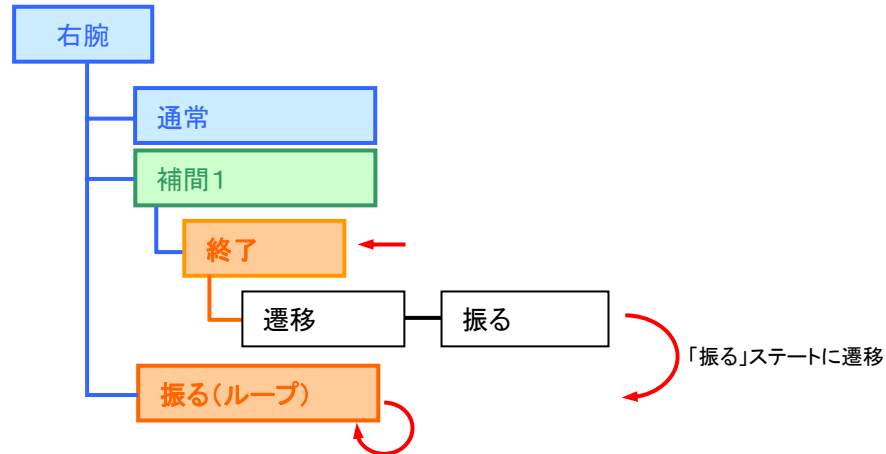
### <通常マスコットデフォルトイベント状態移動イベント[00]が発行された直後>



状態が遷移し、「補間1」のアクションが行われます。

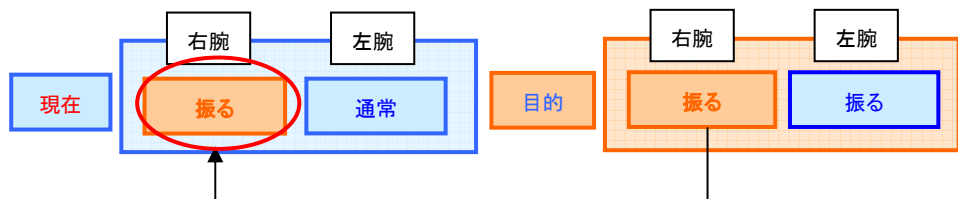
「補間1」のアクションが終了すると同時に「補間1」ステートの「終了イベント」により次の状態に遷移します。

<補間アクションが終了した直後>



補間1のアクションが終了すると終了イベントにより目的状態である「振る」ステートへの遷移が行われます。この遷移が行われた時点で「現在状態」と「目的状態」が同じになり、マチキャラ内部のエンジンは状態の遷移を終了します。

<現在状態と目的状態>



マチキャラは「目的状態」に設定された「振る」ステートへ到達すると「現在状態」に「振る」を設定し、ステートの遷移を終了します。

このような設定を行うことによりマチキャラが以下のように動作します。

【通常状態】



【補間1】



【振る(ループ)】



通常状態から補間動作を経て振る動作を行います。

ここまでの説明で、携帯端末からのイベント発生によりビヘイビアで各モータのステートへの遷移先を指定して、各モータのステートではそれぞれのイベントをハンドリングしながら必要な遷移先を指定することでアクションを連続して再生することがわかりました。

さらに、各イベントでは背景や前景を変更することもできます。モータで間隔イベントを使用することにより、任意のタイミングで背景や前景を変更することができます。

## 9. マチキャラコンテンツ作成におけるガイドライン

ここではマチキャラコンテンツを作成について説明します。

### 9.1. アニメーションの定義

以下にマチキャラデータが動作するアニメーションの定義を行います。

#### 9.1.1. ボーンアニメーション(3D)

モデルデータを構成するボーン(骨)を動作させることにより、アニメーションを行います。

#### 9.1.2. テクスチャアニメーション(3D)

テクスチャが貼られているポリゴンデータをダイナミックに貼りかえることにより、アニメーションを行います。

#### 9.1.3. GIF アニメーション

モデルデータを用いなくてアニメーションの表示を行います。

#### 9.1.4. パルーンアニメーション

キャラクタに付属する吹き出しを、画像とテキストを使用してアニメーションの表示を行います。



## 9.2. 3D マチキャラのキャラクタ作成

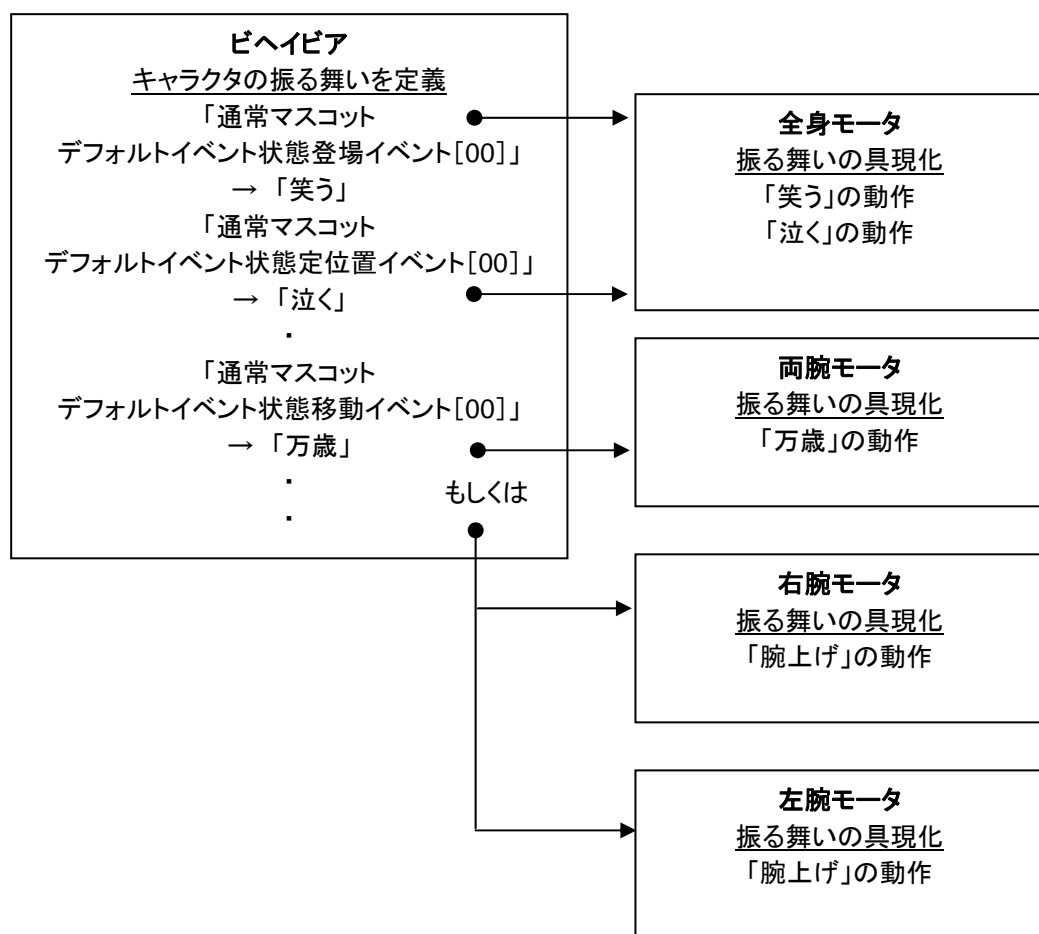
### 9.2.1. モータ(パーツ)の考え方

マチキャラを作成するには、ビヘイビア・モータという概念を使用します。

ビヘイビアとは、モデルの状態を定義し、モータとは、ビヘイビアで定義した振る舞いを具現化する動作を定義するものとなります。

例えば、「通常マスコットデフォルトイベント状態移動イベント[00]」が「万歳」と関連付けることがビヘイビアにあたり、「万歳」という動作がモータに相当します。この「万歳」という動作は「両腕を上げる」という動作により具現化されますが、1つのモータで「両腕」を、また、2つのモータでそれぞれ「左腕」「右腕」を扱うことができます。複数のモータを扱うことをマルチモータと呼びます。

このように1つのモータの動作を1つのパーツとして捉えることができます。



### 9.2.2. アクション

モータで実現するモデルの動作をアクションと呼びます。

マチキャラは、携帯端末の状態等によりさまざまなアクションを行いますが、そのアクションのさせ方により、モータ(パーツ)の捉え方が変わってきます。

全身を扱うアクションのみのマチキャラは、「腕」「足」といった部品のアクション合成の必要がないため、全身を1つのモータとして扱うことになります。

また、全身を扱うアクションも、「腕」「足」といった部品のアクションも行うマチキャラは、それぞれアクション合成が必要になるため、複数のモータに分けて扱うこととなります。



左図のように、上げた両腕を1つのパーツとして扱うか、左腕・右腕を別々のパーツとして扱うか、でモータの構成が変わります。

### 9.2.3. パーツ分け

モデルに対するアクションは、全身のアクションとして作成する場合や、ある意味を持つパーツ単位で作成する場合が存在します。この場合に注意しなければならないのは、ボーンアニメーションの衝突です。例えば、「万歳」アクションを実現する場合に「両腕」を1つのモータとするケース、「左腕」「右腕」として2つのモータとするケースが考えられます。前者の場合、「両腕」モータが動作する上で、他のモータで「両腕」と同じボーンアニメーションが再生されると「ボーンの衝突」が発生し、意図しないアクションとなる可能性があります。

また、後者の場合、「左腕」モータの中に「右腕」モータと同じボーンが、または、「右腕」モータの中に「左腕」モータと同じボーンが存在すると同じ結果となります。

下図のように「右腕を上げる」と「左腕を上げる」アクションがあり、同時に再生した場合に、「右腕を上げる」アクションの中に「左腕」ボーンのアクションが存在すると、「左腕」ボーンの衝突が発生します。



全身のアクションのみの作成時は問題ありませんが、パーツ単位のアクションを作成する必要がある場合は、モータ単位で使用するボーンを決定しボーンが衝突しないようにパーツ分けすることを推奨します。

### 9.2.4. アニメーションの競合

各モータ間のボーンの衝突やテクスチャアニメーションの衝突について定義します。

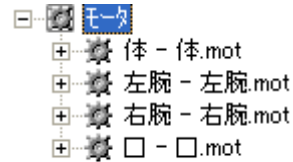
#### 9.2.4.1. ボーンの衝突

マルチモータの場合に、ボーンの衝突が発生することがあります。

基本的には、パーツ分けの時点で各モータ(パーツ)間でのボーンアニメーションが衝突しないようにアクションを定義する必要があります。

ボーンアニメーションの衝突が発生した場合に、キャラクタの動作はモータの優先順位により決定されます。

例) 優先度が体→口順になる。



上図はマチキャラメーカでの設定方法です。  
(詳細はマチキャラメーカ操作説明書を参照してください。)

※ ボーンアニメーションによっては、基本姿勢(アクションしない)ボーンも含まれます。この場合は、それを除いた優先度の高いアニメーションが優先されます。

#### 9.2.4.2. テクスチャアニメーションの衝突

テクスチャアニメーションは、1つのモータで実現します。

テクスチャアニメーションの衝突が発生した場合に、キャラクタの動作はモータの優先順位により決定されます。

※ ボーンアニメーションとは異なり、テクスチャアニメーションは、その位置に関係なく衝突が発生します。例えば、口パクと目パチ等

### 9.3. 2D マチキャラのキャラクタ作成

2Dアニメーションは、3Dアニメーションと異なり、モデルデータ、アクションデータ、テクスチャデータおよびエフェクトデータを使用しません。使用するのは GIF データのみです。

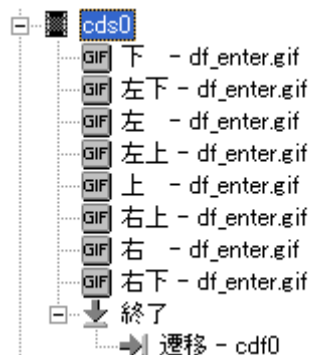
#### 9.3.1. 2D アニメーションの制御

2D アニメーションにおいて、モータで制御する方法を以下に示します。

基本的には 2D アニメーションも 3D アニメーションと同様にモータを設定します。異なるのは、3D アニメーションがアクションファイルを指定するのに対し、2D アニメーションではアニメーション GIF ファイルを指定する点です。また、3D アニメーションではエンジン側でキャラクタの向きを進行方向に合わせて表示するためアクションファイルは1つのみ設定しますが、2D アニメーションではエンジン側でキャラクタの向きを変更できないため、進行方向によって8種類のファイルを設定することができます。8種類全て異なるファイルにする必要はありませんが、その場合でも、モータには8つのファイルを設定する必要があります。同じファイルを複数設定することは可能です。なお、移動イベント以外で使用する場合であっても、それぞれの進行方向について 8 つのファイルを設定する必要があります。どの方向についても同じファイルを利用したい場合は、同じファイルを 8 方向に設定してください。

また、2D アニメーションでは最大アニメーション時間を設定する必要があります。最大アニメーション時間は1ms～10000ms の範囲で1ミリ秒単位で設定することが可能です。最大アニメーション時間が設定されていない場合は1000ms とします。

例)



上図はマチキャラメカでの設定方法です。  
(詳細はマチキャラメカ操作説明書を参照してください。)

※ 2D データでは間隔(TIME)は使用できません。

- ✓ マチキャラ Ver.1.0 のコンテンツを作成する場合は、全ての状態のうち最大となるアニメーション時間を全状態に設定します。また各状態で使用するアニメーションファイルは全て同じアニメーション時間となるように作成する必要があります。

## 9.4. キャラクタの動作設定

### 9.4.1. イベント設定

マチキャラの動きをイベント文字列で制御します。マチキャラメーカーで作成する場合はイベント文字列を意識する必要はありません。

- ✓ イベント文字列については「ビヘイビアとモータの詳細」の【実行可能なイベントコード】を参照してください。

#### 9.4.1.1. 時間指定イベント

携帯端末に登録されているスケジュールおよび携帯端末所有者情報の誕生日になった時に発生するイベントです。それぞれのマスコット状態に外的要因イベントおよび内的要因イベントを設定することができます。

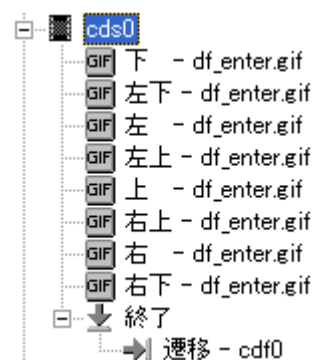
#### 9.4.1.2. 外的要因イベント

携帯端末からの要求により発行されるイベントです。デフォルトイベント、不在着信イベント、未読メールありイベント、メール受信イベント、通話時間イベント、メーカーイベントおよび隠しマスコットイベントがあります。デフォルトイベント、不在着信イベント、未読メールありイベント、通話時間イベントおよびメーカーイベントが実行中の場合、それぞれのイベント状態であるといいます。不在着信イベント中は不在着信イベント状態となります。

#### 9.4.1.3. 内的要因イベント

キャラクタ演出エンジン内で自動発行されるイベントです。登場イベント、移動イベント、定位置イベントおよび癖イベントがあります。

例)



上図はマチキャラメーカーでの設定方法です。

(詳細はマチキャラメーカー操作説明書を参照してください。)

- ✓ ビヘイビア・モータの記述方法詳細はスクリプトリファレンスを参照してください。

### 9.4.2. イベント状態の解除

マチキャラでは、イベント状態を解除するためのイベントを設定する必要はありません。

### 9.4.3. イベントの実行制限

マチキャラでは各イベント毎に実行制限を設定することができます。イベントの実行制限には使用禁止期間、実行比率、重畳シーン別イベント制御の3種類があります。

#### 9.4.3.1. 使用禁止期間

イベント毎に使用禁止期間を設定できます。使用禁止期間は、使用を禁止する期間を秒

単位で設定するか、使用解除する年月日時を指定します。使用禁止期間を設定しないイベントは常時有効となります。

#### 9.4.3.2. 実行比率

実行比率はイベントの重さを任意の数値で指定します。指定されない場合は「10」とみなします。各イベントの重さを現在実行可能な全イベントの重さを合計した値で割った値がそのイベントの実行比率になります。

#### 9.4.3.3. 重畳シーン別イベント制御

重畳シーン別イベント制御は重畳シーン別にイベントを有効または無効に設定します。設定可能な重畳シーンは以下の9種類です。

- ◆ 待ち受け
- ◆ ブラウザ
- ◆ メール
- ◆ テレビ電話
- ◆ メニュー
- ◆ メール受信確認
- ◆ iコンシェル TOP
- ◆ 音声入力モード
- ◆ その他

また、未読メールありイベントは未読メール数はコンテンツで設定した未読メール数より少ない場合は実行されません。同様に、通話時間がコンテンツで設定された通話時間より少ない場合は通話時間イベントは実行されません。

## 9.5. キャラクタの移動

マチキャラでは、キャラクタが携帯端末画面上を移動する動作を設定できます。  
キャラクタの移動は大きく分けて以下の2種類あります。

### 9.5.1. 自動移動

マチキャラ内部のエンジンでランダムに移動目的地点を設定し、移動します。

### 9.5.2. 定型移動

マチキャラコンテンツで指定された移動ルートを移動します。

自動移動の場合は、以下の設定をすることができます。

- ◆ 迂回性能

現在のキャラクタ表示位置から次の目的地点へ向かう際に急な方向転換にならないように、旋回の可否および迂回性能を指定をすることができます。キャラクタの移動スピードとマチキャラデータで指定した迂回性能値によって、2段階の迂回性能があります。キャラクタの移動スピードが速く迂回性能値が小さい場合は大きな迂回に、キャラクタの移動スピードが遅く迂回性能値が大きい場合は小さな迂回になります。

定型移動の場合は、以下で説明するフィットモードと移動モードを設定することができます。

- ◆ フィットモード

定義された移動データを実際の表示可能エリアに反映させる方法です。

- ◇ ストレッチ

縦・横とも表示可能エリアにサイズ調整する方法です。デフォルトのサイズ調整方法です。

- ◇ フィット(短辺長)

縦横比率を変えずに、表示可能エリアの短辺に合わせてサイズ調整する方法です。

- ◇ フィット(長辺長)

縦横比率を変えずに、表示可能エリアの長辺に合わせてサイズ調整する方法です。

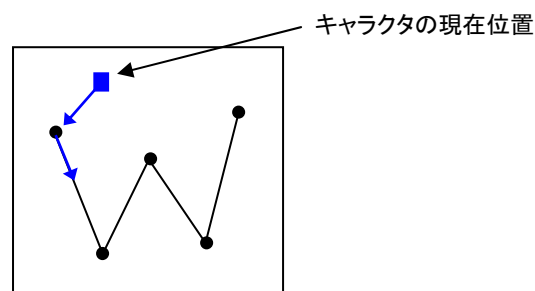
- ◇ サイズ調整なし

縦・横ともサイズ調整をしない方法です。

- ◆ 定型移動モード

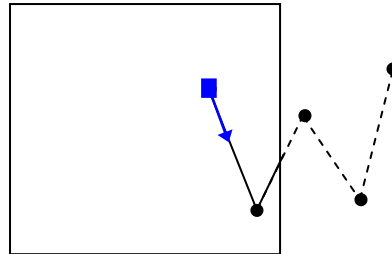
- ◇ 絶対座標

現在のキャラクタ表示位置とは関係なく、マチキャラコンテンツで指定された座標から移動を開始する方法です。現在の表示位置から移動開始位置までキャラクタが移動してから定型移動を開始します。



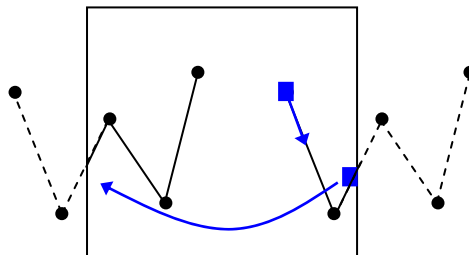
## ◇ 相対ノンエフェクト

現在のキャラクタ表示位置から、定型移動を開始します。移動ルートが携帯端末画面の外へ出る場合は、キャラクタは画面外のルートを移動しているものとし、キャラクタは表示されません。



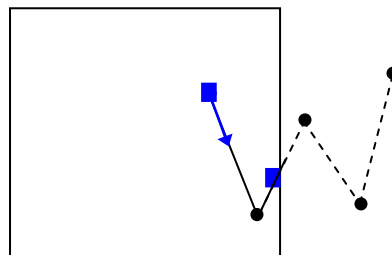
## ◇ 相対エリアループ

現在のキャラクタ表示位置から、定型移動を開始します。移動ルートが携帯端末画面の外へ出る場合は、画面反対側からキャラクタが続きのルートを移動します。



## ◇ 相対エリアエンド

現在のキャラクタ表示位置から、定型移動を開始します。移動ルートが携帯端末画面の外へ出る場合は、画面端で定型移動を終了します。



定型移動、移動移動共通で以下の設定をすることができます。

## ◆ スピード

キャラクタの移動スピードを設定します。スピードを0に設定するとキャラクタはそこで移動動作を行います。

## ◆ 迂回

マチキャラではメニューのフォーカス部分など、キャラクタを表示したくないエリアを移動禁止エリアとし、携帯端末により移動禁止エリアとして設定されたエリアにはキャラクタを表示しません。キャラクタの移動経路が移動禁止エリアと重なる場合の動作について設定することができます。

迂回ありを設定すると、キャラクタは移動禁止エリアを迂回して表示されます。迂回なしを設定すると、キャラクタは移動禁止エリア内を移動ルートどおりに移動し、



移動禁止エリアと重なるエリアではキャラクタは表示されません。

- ✓ 移動禁止エリアは各携帯端末メーカーが設定します。

## 9.6. バルーン表示

マチキャラでは、キャラクタと併せてバルーン表示することができます。  
バルーンは吹き出しのように画像とテキストを設定することもできますし、感情表現（例えばハート）のように画像のみを設定することができます。バルーンを表示しないこともできます。

＜画像とテキストを設定した場合＞



＜画像のみを設定した場合＞



バルーン表現は、複数の画像を使用してアニメーションのようにすることもできます。その場合、各画像に表示時間をミリ秒単位で設定する必要があります。

＜複数の画像を使用した場合＞



バルーンに表示するテキストは画像毎に設定することができます。複数の画像を使用する場合と同様の設定をすることで、1つのテキストを複数回に分けて表示するような表現をすることもできます。

＜テキストを複数回に分けて表示した場合＞



バルーンに表示するテキストは1文字毎に表示する色を設定することができます。

- ✓ 1つのマスコットで設定できる文字色数には制限があります。
- ✓ メッセージの文字数が表示可能文字数を超える場合は、超過分は表示しません。表示可能文字数は以下のとおりです。

ターゲット VRAM エリア	QCIF サイズ	QVGA サイズ	VGA サイズ	XGA サイズ
バルーンエリア サイズ	48 × 16pixel	96 × 32pixel	192 × 64pixel	288 × 96pixel
文字サイズ 12dot	3.5 文字 × 1 行 *1	7 文字 × 2 行	14 文字 × 4 行	21 文字 × 6 行 *2
文字サイズ 18dot	非対応	非対応	非対応	14 文字 × 4 行 *2
文字サイズ 24dot	非対応	3.5 文字 × 1 行 *1	7 文字 × 2 行	非対応
文字サイズ 36dot	非対応	非対応	非対応	7 文字 × 2 行 *2
文字サイズ 48dot	非対応	非対応	3.5 文字 × 1 行 *2	非対応
文字サイズ 72dot	非対応	非対応	非対応	3.5 文字 × 1 行 *2

※V3.0 以降は全角文字を 1 文字、半角文字を 0.5 文字とします。

※V2.0 は全角文字、半角文字とも1文字とします。

\*1 V2.0 の場合は 3 文字 × 1 行になります。

\*2 V3.0 以降のみ対応しています。

- ✓ ターゲット VRAM エリアに合ったバルーンエリアサイズのバルーン画像が設定されていない場合は、バルーンは表示されません。

バルーンの設定はイベントおよび着せ替え表現で設定できます。表示すべきマスコット状態でイベントと着せ替え表現のどちらにもバルーンが設定されていた場合、着せ替え表現のバルーン設定が優先されます。

### 9.6.1. 文字列置換機能

マチキャラデータ V3.0 以降ではバルーンテキストの文字列置換機能を使用することができます。

文字列置換機能はバルーンテキストに登録されている置換コードを設定します。マチキャラデータを携帯端末で実行する際に携帯端末で置換コードを該当する文字列に置き換えて表示します。

文字列置換機能に対応したバルーンテキストを設定する場合は、置換コードを含まない文字列と置換コードを含む文字列の両方を1つのバルーンテキストに設定します。文字列置換機能に対応していない機種では、置換コードを含まない文字列を表示し、文字列置換機能に対応した機種では、置換コードを含む文字列を表示します。

文字列置換機能に対応したバルーンテキストの例は以下のようになります。

＜バルーンテキスト＞

「今日は何ようび？／今日は\* \*ようびだよ」

\* \*:置換コード(例では「曜日」とします。)

＜文字列置換に対応していない機種での表示＞



＜文字列置換に対応している機種での表示＞



\* \* の位置にシステム時刻から算出した曜日表示します。

- ✓ 利用可能な置換コードは、マチキャラメーカーでご確認ください。
- ✓ 置換される文字が1行に収まらない場合、「オート改行あり」に設定されていると自動的に改行が入り次の行に続きが表示され、「オート改行なし」に設定されていると改行が入らず、1行で表示できるところまで表示されます。
- ✓ 文字列置換によりバルーンに表示できる文字数を超えてしまう場合は表示できるところまで表示します。

## 9.7. ポップアップ表示

マチキャラでは、ターゲット VRAM エリアが QVGA、VGA および XGA の場合で、アラーム、ダイアログ、i コンシェル TOP ショートカットフォーカスおよび i コンシェルイベントを実行する場合にポップアップを表示します。

マチキャラコンテンツでは、ポップアップイメージとポップアップに表示する文字色を設定することができます(i モード対応絵文字については、設定された文字色ではなくデフォルト色の表示となります)。



- ✓ ポップアップの表示タイミングおよびポップアップに表示するテキストは移動機により指定されます。
- ✓ ポップアップイメージのサイズはターゲット VRAM エリアによって決められています。各ターゲット VRAM エリアで利用できるポップアップイメージのサイズは以下のとおりです。

ターゲット VRAM エリア	QVGA	VGA	XGA
ポップアップイメージサイズ	228 × 186pixel	457 × 372pixel	684 × 558pixel

- ※ マチキャラメーカーでは最低 1 つのポップアップイメージが設定されていればマチキャラを作成することができますが、移動機にとって必要なサイズのポップアップイメージが存在しないマチキャラは、保存することはできますが、設定することができません。したがって、マチキャラ Ver.3.0 以降対応機種全てに設定可能なマチキャラを作成する場合は、全てのサイズのポップアップイメージを設定しておく必要があります。

## 9.8. 着せ替え表現

マチキャラでは動作はそのまま外見のみを変更する「着せ替え」を設定することができます。着せ替えには基本・時間反映・隠しマスコットの3種類があります。着せ替え表現は設定の順により優先順位が決定します。先に設定された着せ替え条件が優先順位が高く、後に設定された着せ替え条件は優先順位が低くなります。

なお、着せ替え表現にはバルーンの設定もできます。着せ替え表現で設定されたバルーン設定はイベントで設定されたバルーン設定より優先されます。

✓ 1つのマチキャラデータファイルに設定できるマスコットには制限があります。

### 9.8.1. 基本

マチキャラでデフォルトで表示される着せ替えデータです。時間反映または隠しマスコットが適用されると表示が切り替わります。

着せ替えデータを設定します。着せ替えの条件は設定できません。

### 9.8.2. 時間反映表現

マスコットデータに設定された条件により一時的に外見を変更して演出する表現です。時間反映表現は、その時間反映表現をいつ適応するかを指定する時間反映条件と、どのような表現をするかを指定する時間反映データを設定することにより実現します。

時間反映条件には以下の指定ができます。

#### (1)西暦

形式「YYYY」で指定します。指定しない場合は「0000」を設定します。

#### (2)月

形式「MM」で指定します。指定しない場合は「00」を設定します。

#### (3)日

形式「DD」で指定します。指定しない場合は「00」を設定します。

#### (4)季節

識別値で指定します。指定しない場合は「0」を設定します。

識別値

月	1	2	3	4	5	6	7	8	9	10	11	12
季節	冬	冬	春	春	春	夏	夏	夏	秋	秋	秋	冬
識別値	4	4	1	1	1	2	2	2	3	3	3	4

#### (5)曜日

識別値で指定します。指定しない場合は「0」を設定します。

識別値

曜日	日曜日	月曜日	火曜日	水曜日	木曜日	金曜日	土曜日
識別値	1	2	3	4	5	6	7

#### (6)適用開始時刻・適用終了時刻

形式「hhmm-hhmm」で指定します。指定しない場合は「0000-0000」を設定します。

それぞれ単独または組み合わせて設定が可能です。時間反映条件は省略できません。時間反映条件の全てが省略されている場合は、エラーとなります。

### 9.8.3. 隠しマスコット

隠された外見のマスコットです。隠しマスコットイベントに設定します。携帯端末からエンジ

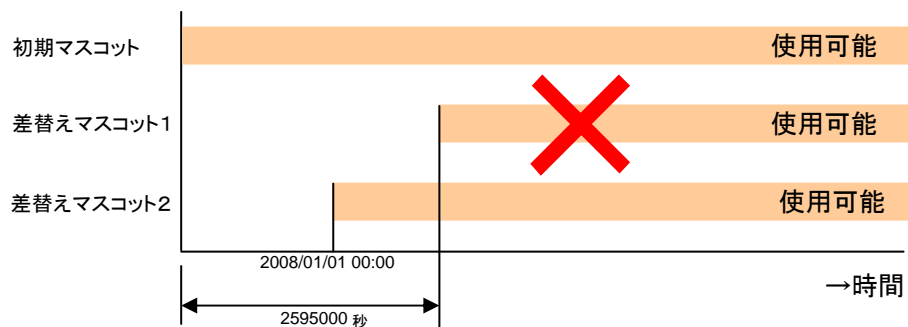
ンに任意のイベントコードを送信した場合に実行されます。イベントコードおよびイベント発行タイミングは携帯端末メーカーが取り決めます。  
着せ替えデータを設定します。着せ替えの条件は設定できません。

## 9.9. マスコット差替え

マチキャラではマスコットデータを複数保持することができます。各マスコットデータに使用禁止期間、使用解除年月日時、積算通話時間、メール送信数およびメール受信数を設定することにより順次表示されるマスコットが更新されます。使用条件は AND 条件または OR 条件を設定します。使用条件は全て設定する必要はありません。条件に含めない場合は「0」を設定することにより条件から除外することができます。

ロード直後に表示可能マスコットが存在しないことを防ぐため、マスコットデータのうち1番目に設定されたマスコットデータを初期マスコットとし使用条件を指定なしで作成します。

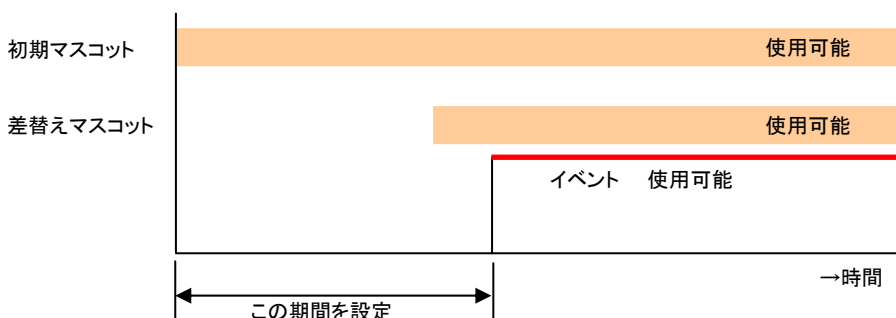
マスコットデータは設定された順に使用可能になるように設定する必要があります。例えば、2番目のマスコットデータの使用禁止期間が2595000秒(30日)、3番目のマスコットデータの使用解除年月日時が2008年01月01日00時に設定されたコンテンツデータを2007年12月02日以降にロードされた場合下図のようになり、正しく動作しません。(自動実行では差替えマスコット1が使用されません。)



上のような事象を防ぐため、マスコットの使用条件を設定する場合は、設定する項目を統一することを推奨します。

使用可能マスコットの一覧からマスコットを選択することにより、マスコット差替えを行わず、選択したマスコットを継続して使用することもできます。マスコットを選択した場合でも、ユーザによるリセット操作を行うことにより、マスコットを指定しないでロードした場合のマスコットを表示します。この場合は、使用条件によりマスコットの差し替えが発生いたします。マスコットを指定してロードする機能の実装／未実装および実装方法は携帯端末メーカーが取り決めます。

差し替えマスコットのイベントに使用禁止期間を設定する場合は、マチキャラコンテンツデータが最初に使用されてから使用可能になるまでの累積使用期間を設定します。該当マスコットに差し替えられてからの期間ではありません。





## 9.10. まとめ

マチキャラを作成するときは以下のガイダンスを基に作成します。

1. マスコットの構成を決める
2. 各マスコットの動作を決める
3. 各マスコットの着せ替え表現を決める
4. 各マスコットのバルーン表現を決める

マスコットの構成ではマスコットデータの数およびマスコットデータをいつ使用可能にするかを決めます。

更に各マスコットでの動作を作成するときには以下のようなガイダンスを基に作成します。

1. アクションの動作を決める
2. ループするアクションでは状態を遷移する際に補間アクションを考える
3. 各アクションで動作させるパーツを決定する
4. パーツ毎に遷移図を作成する
5. 遷移図を基にパーツ毎に状態を洗い出す
6. ビヘイビアとモータの概念を元にイベントとアクションを設定する
7. 各イベントの実行条件を決める

その他、ビヘイビア、モータを扱うにあたり注意しなければならない点は以下のとおりです。

### 1. モータの優先順位

各モータのステートにおいてキャラクタのそれぞれのパーツを制御しますが、各パーツでの制御の際に他のパーツを操作するような動作がモータ間を跨いで同時に発生した場合は、先に設定しているモータでの設定が有効になります。(ボーンの衝突)

### 2. 背景・前景を連続で設定した場合

背景を連続で設定した場合は毎回描画が行われませんので、画面には最後の背景が設定されます。

### 3. ボーンの衝突

マルチモータの場合はモータ間のボーンアニメーションが衝突しないようにアクションの定義とモータのパーツ分けをする必要があります。ボーンアニメーションの衝突が発生した場合に、キャラクタの動作はモータの優先順位により決定されます。

### 4. テクスチャアニメーション

モータを跨いでテクスチャアニメーションの設定を行う場合はテクスチャアニメーションが衝突(同時に再生)しないように注意してください。衝突した(同時に再生が行われた)場合にはモータの優先順位に従って動作します。

### 5. 作成するアクションの長さ

キャラクタのアクションを作成する際には、同時に行うパーツ毎のアクションの長さ(時間)は同じ長さになるように作成することをお勧めします。これは、例えば両腕を同時に上げる場合に右腕と左腕のパーツでそれぞれの腕を上げるアクションの時間が異なるとその次の状態遷移を行ったときに各パーツがずれて再生されてしまうからです。

### 6. 作成するモデルの表現

キャラクタを作成する際には、使用する表現によりポリゴン属性を設定する必要があります。キャラクタに平行光源または環境マッピングを使用する場合はポリゴン属性に平行光源を設定する必要があります。また、キャラクタにブレンディングを使用

する場合はキャラクタを作成する際にポリゴン属性を設定します。この場合、携帯端末に表示する際は、キャラクタがマスコットエリアとブレンドされた画像が携帯端末に表示されます。キャラクタと携帯端末画像はブレンドされないので注意してください。

## 7. フレームレート

携帯端末上ではフレームレートは通常8FPS程度です。マチキャラコンテンツを作成する場合はフレームレートを8FPS程度に設定して描画状態を確認してください。(マチキャラメーカーのプレビューウィンドウではフレームレートのデフォルト設定は8FPSに設定されています。)

### マスコットエリア …

マチキャラでキャラクタを表示するエリア。

ターゲットVRAMエリアの大きさとマスコットエリアサイズ設定コマンドの設定により、マスコットエリアサイズは以下ようになります。

ターゲットVRAM エリア短辺長		QCIF サイズ	QVGA サイズ	VGA サイズ	XGA サイズ
		96～239 pixel	240～479 pixel	480～719 pixel	720 ～ 1280 pixel
マ ス コ ッ ト エ リ ア	Sサイズ	32×32 pixel	48×48 pixel	96×96 pixel	144×144 pixel
	Mサイズ	48×48 pixel	64×64 pixel	128×128 pixel	192×192 pixel
	Lサイズ	64×64 pixel	96×96 pixel	192×192 pixel	288×288 pixel
	LLサイズ	96×96 pixel	128×128 pixel	256×256 pixel	384×384 pixel

※ アラーム、ダイアログ、iコンシェル TOP ショートカットフォーカス、iコンシェル、iコンシェル TOP イベント時は以下の重畳シーンではマスコットエリアは M サイズとなります。

イベント	マスコットエリア制御対象重畳シーン
iコンシェルイベント	待ち受け画面 iコンシェル TOP 画面
アラームイベント	待ち受け画面 iコンシェル TOP 画面
ダイアログイベント	音声入力画面
iコンシェル TOP ショートカットフォーカスイベント	待ち受け画面
iコンシェル TOP	iコンシェル TOP 画面

### ターゲットVRAMエリア …

移動機側でマチキャラ画像の描画を許可する矩形エリア。

ターゲットVRAMエリアの大きさは以下の3種類に分けられます。

種類	QCIF サイズ	QVGA サイズ	VGA サイズ	XGA サイズ
ターゲットVRAM エリア短辺長	96～239 pixel	240～479 pixel	480～719 pixel	720 ～ 1280 pixel

## スクリプトリファレンス

本章では以下のスクリプト文法について説明します。

- ◆ ビヘイビアファイル
- ◆ モータファイル
- ◆ キャラクターメイクファイル
- ◆ バルーンデータファイル
- ◆ 定型移動データファイル
- ◆ ポップアップデータファイル

## S.1. 構文表記法

本章では、文法を説明するために使用している表記法について説明します。スクリプトの文法は以下の形式で定義します。

### S.1.1. 構文のルール

まずその構文の名称があり、それにイコール(=)が続いています。イコールの右辺は、構文のルールを定義しています。小なり(<)と大なり(>)で囲まれた語は特定の構文の要素を示しています。

次の例は、<A>という構文のルールを説明しています。

```
<A> = <B> | <C>
```

### S.1.2. 見出語

太字で示された語は、見出語を表しています。スクリプト中でもそのままの語が使用されます。次の例では、“play”という見出語はスクリプト中にそのままの語を記述することを示しています。

```
play { [ command... ] }
```

### S.1.3. 選択肢

選択肢を構成する垂直バー(|)は、垂直バーで区切られたうちのいずれかの値を取る、という選択肢を表しています。選択肢が一部分だけを構成する場合は、左カッコ(と右カッコ)で囲んでいます。まず最初の例では、<A>は<x>または<y>であることを示しています。また、<B>は「<x> 1 <y>」、「<x> 2 <y>」、または「<x> 3 <y>」のいずれかになります。

```
<A> = <x> | <y>
<B> = <x> ( 1 | 2 | 3 ) <y>
```

### S.1.4 繰り返し

繰り返しは次のように表現しています。左カギカッコ[と右カギカッコ]で囲まれた部分は、省略可能であることを示しています。a [ a... ] という表現は、a を 1 回以上繰り返すことを示しています。ただし a を 2 回以上繰り返す場合は空白で区切ります。例外がある場合には随時補足してあります。

次に上記の例を示します。

```
<A> = x [ y ]
<B> = x [ x... ]
```

この例では、それぞれ次のようになります。

```
<A> は「x」または「x y」のいずれか
<B> は「x」、「x x」、「x x x」など
```

また、以下の文字 1 - 文字 2 という表現は、文字 1 から文字 2 までの範囲に含まれる文字のいずれかを示しています。例えば次の例は<A>も<B>も a、b、c、d、e、f のいずれかの 1 文字となることを示しています。

```
<A> = a - f
<B> = ( a | b | c | d | e | f )
```

## S.2. 字句要素

本章では、構文を構成する字句要素について説明します。

### S.2.1. 文字集合

ビヘイビア・モータスクリプト内では、以下の文字集合を使用することができます。

#### S.2.1.1. アルファベット

大文字と小文字のアルファベット52文字

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

アルファベットの大文字と小文字は区別されます。

#### S.2.1.2. 数字

10進数字10個

0 1 2 3 4 5 6 7 8 9

#### S.2.1.3. 図形文字

図形文字として以下の文字が使用可能です。

文字	意 味	文字	意 味
!	感嘆符	~	チルダ
#	シャープ	'	一重引用符
\$	ドル記号	^	べき乗
&	アンパサント	`	コマンド実行
(	左カッコ	*	アスタリスク
)	右カッコ	¥	バックスラッシュ
-	マイナス		垂直バー
+	プラス	;	セミコロン
=	イコール	:	コロン
[	左カギカッコ	"	二重引用符
]	右カギカッコ	,	カンマ
{	左中カッコ	<	小なり
}	右中カッコ	>	大なり
@	アットマーク	/	スラッシュ
.	ピリオド	?	クエスチョン

#### S.2.1.4. 特殊文字

以下の文字は Windows の制限により、ファイル名に使用することはできません。

文字	意 味	文字	意 味
*	アスタリスク	"	二重引用符
¥	バックスラッシュ	,	カンマ
	垂直バー	<	小なり
;	セミコロン	>	大なり
:	コロン	/	スラッシュ
?	クエスチョン		

### S.2.1.5. 水平タブ

スクリプト中に現れる水平タブは、文字の区切りとして処理され、水平タブは無視されます。

### S.2.1.6. 改行文字

スクリプト中に現れる改行文字は、文字の区切りとして処理され、改行は無視されます。ただし、以下のケースは別扱いとなります。

コメントでは、改行文字がコメントの終わりを表します。

バルーンテキストを除く二重引用符に囲まれた[文字列定数](#)内に改行が現れた場合はシンタックスエラーとなります。

### S.2.1.7. コメント

[文字列定数](#)以外の中に感嘆符(!)が現れた場合、それ以降の文字列はコメントとして、スクリプト上は空白と同等に処理されます。コメントには文字数の制限はありません。感嘆符から次の改行文字までがコメントとなります。

<コメント> = ! <任意の文字列> <改行文字>

但し、コメントはマチキャラメーカに取り込まれませんので、マチキャラメーカで一度編集を行うと破棄されます。

### S.2.1.8. トークン

スクリプト中に現れる以下の文字を区切りとしてトークンを切り出します。

文字	意味	文字	意味
	空白	{	左中カッコ
¥n	改行	}	右中カッコ
¥r	復帰	:	コロン
¥t	水平タブ	;	セミコロン
!	感嘆符		

### S.2.1.9. トークンの回避

ファイル名を表す場合には、二重引用符(")で対象文字を囲むことにより特殊文字(改行・復帰、:(コロン)、;(セミコロン))を除くトークンを無視させることができます。

### S.2.1.10. その他

文字列として漢字の使用は可能ですがファイル名において空白文字を含む指定を行うと、上記トークンのルールにより、別の文字列として認識されます。

setfg 背景 その1.bmp

上記は「setfg」、「背景」、「その1.bmp」と3つの文字列として認識します。ファイル名として「背景 その1.bmp」と認識させる必要がある場合は、setfg "背景 その1.bmp" のように二重引用符で囲む必要があります。

### S.3. 構文規則

この章では以下のスクリプトについて各セクション毎に説明します。

- ◆ ビヘイビアファイル
- ◆ モータファイル
- ◆ キャラクターメイクファイル
- ◆ バルーンデータファイル
- ◆ 定型移動データファイル
- ◆ ポップアップデータファイル

### S.3.1. ビヘイビアファイル

ビヘイビアファイルでは、以降に説明する各セクション毎の定義によって、イベントの受取り、モータの状態の一括変更、目的状態の設定<sup>1</sup>を行い、マチキャラの振る舞いを決定します。テキストファイルとして作成します(拡張子は.bhv)。ビヘイビアファイル名に**特殊文字**を含めることはできません。ビヘイビアファイルの漢字コードはシフトJISで作成します。

以降、ビヘイビアファイル内でのセクション定義について説明します。  
ビヘイビアを定義するための構文は以下のように定義します。

```
<behavior-name> {
    [ model { <model-file> } ]
    [ motor { <motor-file> [ <motor-file>... ] } ]
    [ texture { <texture-file> [ <texture-file>... ] } ]
    [ map { <map-file> } ]
    [ move { <move-file> } ]
    [ balloon { <balloon-file> } ]
    [ dress { <condition> <action-file> [ <balloon> ]
              [ <condition> <action-file> [ <balloon> ] ... ] } ]
    event-id1 {
        [ event-id2 {
            <event-id3> { [ <command>... ] }
            [ <event-id3> { [ <command>... ] } ... ]
        } ]
        [ event-id2 {
            <event-id3> { [ <command>... ] }
            [ <event-id3> { [ <command>... ] } ... ]
        } ... ]
    }
    [ event-id1 {
        [ event-id2 {
            <event-id3> { [ <command>... ] }
            [ <event-id3> { [ <command>... ] } ... ]
        } ]
        [ event-id2 {
            <event-id3> { [ <command>... ] }
            [ <event-id3> { [ <command>... ] } ... ]
        } ... ]
    } ... ]
}
```

☆ <behavior-name>(ビヘイビア名)

ビヘイビア名を定義する文字列です。

- ✓ ビヘイビアファイルには1つのビヘイビアのみ定義できます。
- ✓ **特殊文字**、**トークン**は使用できません。

<sup>1</sup> ビヘイビアとモータを使用してマチキャラの状態を遷移させることがスクリプトの役割ですが、スクリプトではマチキャラの状態について「現在状態」と「最終的な状態(目的状態)」を設定することができます。「現在状態」を設定するコマンド(set)ではマチキャラの状態を set コマンドで設定された状態に変更します(指定した動作が即実行されます)。また、「最終的な状態(目的状態)」を設定するコマンド(play)では現在のマチキャラの状態は変更せず、play コマンドによって設定された状態をマチキャラに設定するまで状態を遷移し続けます。(設定された動作までの間の動作を順次実行させることができます。)



- ◇ model motor texture map move balloon dress event（セクション）  
以降に説明する各セクションには定義する順番に指定があります。  
以下の点に考慮して定義してください。他はどの順番でも構いません。
  - ✓ motor セクションは event セクションより前に定義すること。
  - ✓ balloon セクションは dress セクションより前に定義すること。
  
- ◇ 重複定義  
各セクションは重複して定義できません。

### S.3.1.1. model セクション

model セクションでは、マチキャラを作成する際に使用するモデルファイル名を指定します。  
3D マチキャラを作成する場合にはこのセクションにモデルファイル名の指定を行います。  
2D マチキャラを作成する場合には指定しません。

model セクションは以下のように定義されます。

```
model { <model-file> }
```

◇ model セクション

✓ model セクションの複数定義はできません。

◇ <model-file> (モデルファイル名)

モデルファイル名(.bac または.mbac)を指定します。

- ✓ 存在するモデルファイルである必要があります。
- ✓ 複数のモデルファイルは定義できません。
- ✓ 漢字ファイル名は使用可能ですが、ファイル名に[特殊文字](#)は使用できません。
- ✓ モデルファイル名に[特殊文字](#)を除く[トークン](#)を含める場合は二重引用符で囲む必要があります。
- ✓ model セクションの定義を行った場合はモデルファイルの指定は必須となります。

例)モデルファイルとして「figure.bac」を使用する場合。

```
model { figure.bac }
```

例)モデルファイルとして「figure 01.bac」を使用する場合。

```
model { "figure 01.bac" }
```

ファイル名に空白を含める場合は、二重引用符で囲む必要があります

### S.3.1.2. motor セクション

motor セクションでは、使用するモータとその順番をモータファイル名を指定することで定義します。以降に説明する event セクションで定義するモータはこのセクションで定義しておく必要があります。

motor セクションは以下のように定義されます。

```
motor { <motor-file> [ <motor-file>... ] }
```

◇ motor セクション

- ✓ motor セクションの複数定義はできません。

◇ <motor-file> (モータファイル名)

モータファイル名(.mot)を指定します。

- ✓ 存在するモータファイル名である必要があります。
- ✓ 複数のモータファイル名を定義する場合は空白で区切ります。
- ✓ 同一モータファイル名の重複定義はできません。
- ✓ 定義できるモータファイル数は最大16個です。
- ✓ 漢字ファイル名は使用可能ですが、ファイル名に[特殊文字](#)は使用できません。
- ✓ モータファイル名に[特殊文字](#)を除く[トークン](#)を含める場合は二重引用符で囲む必要があります。
- ✓ モータファイル名の定義の順番により、アクションの優先度<sup>2</sup>が決定されます。
- ✓ event セクションよりも前に定義する必要があります。

例)モータファイル名として「motor1.mot」と「motor2.mot」を使用する場合。

```
motor { motor1.mot motor2.mot }
```

例)モータファイル名として「motor 1.mot」を使用する場合。

```
motor { "motor 1.mot" }
```

ファイル名に空白を含める場合は、二重引用符で囲む必要があります

<sup>2</sup> アクションの合成時に同一ボーンへのアクションが発生した場合には、当セクションにて指定されたモータファイルの左から順に優先度が高く再生されます。

### S.3.1.3. texture セクション

texture セクションでは、モデルに適用するテクスチャファイル名 (.gif、.bmp) を指定します。  
使用するモデルに合わせたテクスチャファイル名を指定します。  
2D マチキャラを作成する場合には指定しません。

texture セクションは以下のように定義されます。

```
texture { <texture-file> [ <texture-file>... ] }
```

◇ texture セクション

- ✓ texture セクションの複数定義はできません。
- ✓ texture セクションを定義した場合は model セクションの定義が必須となります。

◇ <texture-file> (テクスチャファイル名)

- ✓ 存在するテクスチャファイル名である必要があります。
- ✓ 複数のテクスチャファイル名を定義する場合は空白で区切ります。
- ✓ 定義できるテクスチャファイル数は最大16個です。
- ✓ 漢字ファイル名は使用可能ですが、ファイル名に**特殊文字**は使用できません。
- ✓ テクスチャファイル名に**特殊文字**を除く**トークン**を含める場合は二重引用符で囲む必要があります。
- ✓ モデルに対するテクスチャの順番を意識して<sup>3</sup>設定する必要があります。

例) テクスチャファイルとして「texture1.gif」と「texture2.gif」を使用する場合。

```
texture { texture1.gif texture2.gif }
```

例) テクスチャファイルとして「texture 1.gif」と「texture 2.gif」を使用する場合。

```
texture { "texture 1.gif" "texture 2.gif" }
```

ファイル名に空白を含める場合は、二重引用符で囲む必要があります

<sup>3</sup> 作成したモデルに設定しているテクスチャ ID の順に定義する必要があります。

#### S.3.1.4. map セクション

map セクションでは、モデルに適用する環境マッピングファイル名 (.gif、.bmp) を指定します。  
使用するモデルに合わせた環境マッピングファイル名を指定します。  
2D マチキャラを作成する場合には指定しません。

map セクションは以下のように定義されます。

```
map { <map-file> }
```

◇ map セクション

- ✓ map セクションの複数定義はできません。
- ✓ map セクションを定義した場合は model セクションの定義が必須となります。

◇ <map -file> (環境マッピングファイル名)

- ✓ 存在する環境マッピングファイル名である必要があります。
- ✓ 定義できる環境マッピングファイル数は最大1個です。
- ✓ 漢字ファイル名は使用可能ですが、ファイル名に**特殊文字**は使用できません。
- ✓ 環境マッピングファイル名に**特殊文字**を除く**トークン**を含める場合は二重引用符で囲む必要があります。
- ✓ map セクションの定義を行った場合は環境マッピングファイルの指定は必須となります。

例) 環境マッピングファイルとして「map 1.gif」を使用する場合。

```
map { map 1.gif }
```

例) 環境マッピングファイルとして「map 1.gif」を使用する場合。

```
map { "map 1.gif" }
```

ファイル名に空白を含める場合は、二重引用符で囲む必要があります

### S.3.1.5. move セクション

move セクションでは、定型移動データ情報を設定します。

move セクションは以下のように定義されます。

```
move { <move-file> }
```

☆ move セクション

✓ move セクションの複数定義はできません。

☆ < move-file> (定型移動データファイル名)

定型移動データファイル名(.fmf)を指定します。

- ✓ 存在する定型移動データファイル名である必要があります。
- ✓ 定義できる定型移動データファイル数は最大1個です。
- ✓ 漢字ファイル名は使用可能ですが、ファイル名に特殊文字は使用できません。
- ✓ 定型移動データファイル名に特殊文字を除くトークンを含める場合は二重引用符で囲む必要があります。

例) 定型移動データファイルとして「move0.fmf」を使用する場合。

```
move { move0.fmf }
```

例) 定型移動データファイルとして「move 0.fmf」を使用する場合。

```
move { "move 0.fmf" }
```

ファイル名に空白を含める場合は、二重引用符で囲む必要があります

### S.3.1.6. balloon セクション

balloon セクションでは、バルーンデータファイル名を設定します。

balloon セクションは以下のように定義されます。

```
balloon { <balloon-file> }
```

☆ balloon セクション

- ✓ balloon セクションの複数定義はできません。

☆ < balloon-file > (バルーンデータファイル名)

バルーンデータファイル名 (.bln) を指定します。

- ✓ 存在するバルーンデータファイル名である必要があります。
- ✓ 複数のバルーンデータファイルは定義できません。
- ✓ ファイル名に漢字は使用できませんが、[特殊文字](#)は使用できません。
- ✓ バルーンデータファイル名に[特殊文字](#)を除く[トークン](#)を使用する場合は二重引用符で囲む必要があります
- ✓ dress セクションよりも前に定義する必要があります。

例) バルーンデータファイルとして「balloon0.bln」を使用する場合

```
balloon { balloon0.bln }
```

例) バルーンデータファイルとして「balloon 0.bln」を使用する場合

```
balloon { "balloon 0.bln" }
```

ファイル名に空白を使用する場合は、二重引用符で囲む必要があります。

### S.3.1.7. dress セクション

dress セクションでは、時間反映表現情報を設定します。

dress セクションは以下のように定義されます。

```
dress { <condition> <action-file> [ <balloon>];
      [ <condition> <action-file> [ <balloon>]; ]... }
```

#### ◇ dress セクション

- ✓ dress セクションの複数定義はできません。
- ✓ balloon セクションより後に定義する必要があります。

#### ◇ < condition > (適用条件)

時間反映適用条件を指定します。

- ✓ 書式は“yyyymmdd season weekday hhmm-hhmm”です。
- ✓ 指定しない項目は「0000」、「00」および「0」を指定します。
- ✓ 時間反映表現の基本を設定する場合は「default」を指定します。
- ✓ 最低1項目は指定する必要があります。
- ✓ season および weekday の設定値は以下のとおりです。

season 設定値

月	1	2	3	4	5	6	7	8	9	10	11	12
季節	冬	冬	春	春	春	夏	夏	夏	秋	秋	秋	冬
設定値	4	4	1	1	1	2	2	2	3	3	3	4

weekday 設定値

曜日	日曜日	月曜日	火曜日	水曜日	木曜日	金曜日	土曜日
識別値	1	2	3	4	5	6	7

#### ◇ < action-file > (時間反映ファイル)

時間反映ファイル名を指定します。

- ✓ 存在する時間反映ファイル名である必要があります。
- ✓ 定義できる時間反映ファイル数は最大1個です。
- ✓ 漢字ファイル名は使用可能ですが、ファイル名に**特殊文字**は使用できません。
- ✓ 時間反映ファイル名に**特殊文字**を除く**トークン**を含める場合は二重引用符で囲む必要があります。

#### ◇ < balloon > (バルーン情報)

バルーン情報を指定します。

- ✓ 書式は“on/off QCIFbal-name QVGAbal-name VGAbal-name XGAbal-name”です。
- ✓ on/off は0(無効)または1(有効)のいずれかを指定します。
- ✓ QCIFbal-name はターゲットVRAMエリア<sup>4</sup>が QCIF サイズの場合に使用するバルーン名を指定します。
- ✓ QVGAbal-name はターゲットVRAMエリアが QVGA サイズの場合に使用するバルーン名を指定します。

<sup>4</sup> 移動機側でマチキャラ画像の描画を許可する矩形エリアです。  
ターゲット VRAM エリアの大きさは以下の4種類に分けられます。

種類	QCIF サイズ	QVGA サイズ	VGA サイズ	XGA サイズ
ターゲット VRAM エリア短辺長	96～239pixel	240～479pixel	480～719pixel	720～1280 pixel



- ✓ VGBal-name はターゲットVRAMエリアが VGA サイズの場合に使用するバルーン名を指定します。
- ✓ XGBal-name はターゲットVRAMエリアが XGA サイズの場合に使用するバルーン名を指定します。
- ✓ QCIFbal-name、QVGBal-name、VGBal-name および XGBal-name は存在するバルーン名である必要があります。
- ✓ on/off が無効の場合は QCIFbal-name、QVGBal-name、VGBal-name および XGBal-name は省略します。
- ✓ on/off が有効の場合は QCIFbal-name、QVGBal-name、VGBal-name および XGBal-name を省略することはできません。
- ✓ on/off が有効で該当サイズのバルーン画像が存在しない場合は、バルーン非表示とし、該当する項目を-1に設定します。
- ✓ バルーン設定が条件なしの場合はバルーン情報を省略します。

例)適用条件「2009 年夏」、時間反映ファイルとして「dress0.tra」、バルーン名としてターゲット VRAM エリアが QCIF サイズでは「Skumo」、VGA サイズでは「Lkumo」を使用し、QVGA サイズおよび XGA サイズではバルーンを表示しない場合。

```
dress { 20090000 2 0 0000-0000 dress0.tra 1 Skumo -1 Lkumo -1 }
```

例)適用条件「日曜日 10:00～17:00」、時間反映ファイルとして「dress 0.tra」を使用し、バルーンを表示しない場合。

```
dress { 00000000 0 1 1000-1700 "dress 0.tra" 0 }
```

ファイル名に空白を含める場合は、二重引用符で囲む必要があります

### S.3.1.8. event セクション

event セクションでは、背景・前景および、状態を設定する各コマンドを定義することにより、マチキャラの動作または状態を設定します。

event セクションは以下のように定義されます。

```
event-id1 {
  event-id2 {
    event-id3 {
      [ event ( on | off ); ]
      [ setfg <foreground-file> [attribute]; ]
      [ setbg <background-file> [attribute]; ]
      modeloffset <X> <Y>;
      modelscale <scale>;
      [ light <light-id> <X> <Y> <Z> <light-intensity> <ambient-intensity>; ]
      [ fix <fixedmove-index> <fit-mode> <move-mode>; ]
      [ detour <on/off>; ]
      [ turn < turn >; ]
      [ speed <speed>; ]
      [ areasize <size> <time>; ]
      [ conceal <on/off> <action-file> [ <balloon>; ] ]
      [ balloon <on/off> [ <Simage-name> <Mimage-name> <Limage-name> ]; ]
      [ useful <id> <value>; ]
      [ ratio <ratio>; ]
      [ scene <on/off> <scene-no>; ]
      [ unread <unreadmail>; ]
      [ call <time>; ]
      [ ( play { <motor-file>:<state> [ <motor-file>:<state>... ] } |
        set { <motor-file>:<state> [ <motor-file>:<state>... ] }); ]...
    }
  }
}
```

#### ◇ event セクション

- ✓ event セクションの複数定義はできますが同じイベントコードの重複はできません。
- ✓ motor セクションより後に定義する必要があります。
- ✓ 以降に説明する各コマンド  
(event/setfg/setbg/modeloffset/modelscale/light/fix/detour/turn/speed/areasize/conceal/balloon/useful/ratio/scene/unread/call/play/set)の間は;(セミicolon)で区切ります。
- ✓ 各コマンド(event/setfg/setbg/modeloffset/modelscale/light/fix/detour/ turn/speed/areasize/conceal/balloon/useful/ratio/scene/unread/call/play/set)は定義された順番に動作します。
- ✓ コマンド scene、play、set を除き、同一コマンドが複数定義された場合は、最後に定義された内容が有効になります。
- ✓ コマンド fix と turn は同時に設定することはできません。
- ✓ event-id1 は時間指定コードを設定します。  
通常の場合は「default」、時間指定の場合は「schedule\*\*」(\*\*はスケジュール属性コード)と設定します。

- ✓ event-id2 はイベント状態コードを設定します。
- ✓ event-id3 はイベント属性およびインデックスを設定します。
- ✓ 時間指定コード、イベント状態コードイベント属性およびインデックスはイベントコード仕様を参照してください。

◆ イベントの種類

◇ 外的要因イベント(メーカ実装イベント)

- アラームイベント  
スケジュールのアラームをユーザに通知する場合に発生するイベント。  
アクション種別ごとに設定することができます。
- ダイアログイベント  
音声対話機能を利用中に移動機から要求があった場合に発生するイベント。  
アクション種別ごとに設定することができます。
- i コンシェル TOP ショートカットフォーカスイベント  
待ち受け画面上の i コンシェル TOP ショートカットにフォーカスが当たったことをユーザに通知する場合に発生するイベント。  
アクション種別ごとに設定することができます。
- i コンシェルイベント  
移動機から要求があった場合に発生するイベント。  
アクション種別ごとに設定することができます。
- i コンシェル TOP イベント  
i コンシェル TOP のテンプレートに含まれる全画面上で、表示中の画面と連携してユーザに通知する場合に発生するイベント。  
アクション種別ごとに設定することができます。
- 不在着信通知イベント  
不在着信があった場合に発生するイベント。
- 未読メールあり通知イベント  
移動機で保持している未読メール数に変化が生じた場合に移動機からキャラクタ演出エンジンに未読メール数が通知され発生するイベント。
- メール受信通知イベント  
メールの受信があった場合に発生するイベント。「メール」とは「iモードメール」、「SMS」を指します。
- 通話時間イベント  
移動機から通知される音声またはテレビ電話の直近の通話時間により発生するイベント。
- メーカイベント  
携帯端末メーカ固有の文字列イベント。  
イベントコードは10文字固定の「#」を含む 0～9 の半角数字で構成される任意の文字列。イベントコードは携帯端末メーカが取り決めます。
- 隠しマスコットイベント  
隠された外見ステート(着せ替え)に変更する文字列イベント。  
イベントコードは最大8文字の「#」を含む 0～9 の半角数字で構成される任意の文字列。イベントコードは携帯端末メーカが取り決めます。
- 時間指定イベント  
移動機に登録されているスケジュールの日時および所有者情報の誕生日になった場合にマスコットの表示を一時的に一部変更するイベント。それぞれのスケジュール属性に対し設定することができます。
- リセットイベント  
マスコットの動作をリセットするイベント。

- ◇ 内的要因イベント(キャラクタエンジン内で自動発生するイベント)
  - 登場イベント  
マスコットの場面遷移時に発生するイベント。
  - 移動イベント  
マスコットが画面位置を移動する内部イベント。
  - 定位置イベント  
マスコットが画面上の定位置で自動発生する内部イベント。
  - 癖イベント  
キャラクタ演出エンジン側で常に10%の確率で自動発行される内部イベント。

- ◆ イベントコード仕様  
イベントコードの仕様は以下のとおりです。

イベントコード			
ブロック1	ブロック2	ブロック3	ブロック4
時間指定コード	イベント状態コード	イベント属性	インデックス
「D」: 通常 「S**」: 時間指定 「**」にはスケジュール属性コードが入る	「D」: デフォルト状態 「W**」: アラーム 「X**」: ダイアログ 「Y**」: i コンシェル TOP ショートカットフォーカス 「A**」: i コンシェル 「Z**」: i コンシェル TOP 「**」にはアクション種別が入る 「P」: 不在着信 「N」: 未読メールあり 「M」: メール受信 「C」: 通話時間 「[文字列]」: メーカイベント 隠しマスコットイベント	「S」: 登場 「F」: 定位置 「M」: 移動 「H」: 癖	00～99

※インデックスは「00」からの連番である必要があります。

時間指定イベントの「スケジュール属性コード」の仕様は以下のとおりです。

コード	説明	コード	説明
00	無し	11	スポーツ
01	記念日	12	鑑賞
02	端末所有者の誕生日	13	ドライブ
03	デート	14	会議
04	休日	15	趣味
05	プライベート	16	飲み会
06	仕事	17	習い事
07	買い物	18	出張
08	カラオケ	19	病院
09	旅行	20	スケジュールに登録されている誕生日
10	食事	21～99	予備

- ◇ コマンドに関しては、[コマンド詳細 \( event \)](#)を参照ください。

### S.3.1.9. コマンド詳細 ( event )

event において以下に説明する各コマンドを指定することでマチキャラの動作を制御します。

- ◇ event (イベント受付コマンド)
  - on または off を設定することにより、実行中に他のイベントを受け付けるか否かを設定します。
  - ✓ on または off のいずれかを指定します。(デフォルトは on)
- ◇ setfg (前景設定コマンド) / setbg (背景設定コマンド)
  - setfg < foreground-file> [attribute]により前景画像を指定する場合に使用します。
  - setbg < background-file> [attribute]により背景画像を指定する場合に使用します。
  - ✓ 存在するファイルである必要があります。
  - ✓ 指定できる画像ファイルは GIF89a または Bitmap 形式である必要があります。
  - ✓ 漢字ファイル名は使用可能ですが、ファイル名に**特殊文字**は使用できません。
  - ✓ 前景／背景ファイル名に**特殊文字**を除く**トークン**を含める場合は二重引用符で囲む必要があります。
  - ✓ setfg コマンドを定義した場合は前景画像の設定は必須です。
  - ✓ [attribute]には属性として以下の内容が設定可能です。

tile	タイルリング
zoom	ストレッチ
center	センタリング

何も指定しない場合は「属性なし」となります。
  - ✓ 設定されている画像について同じ属性が指定されていなければなりません。
  - ✓ マチキャラでは、前景／背景として設定する画像ファイルの属性は全て「zoom」として扱います。
- ◇ modeloffset (モデル位置設定コマンド)
  - modeloffset <X> <Y> によりモデルの位置を指定する場合に使用します。
  - ピクセル単位で指定します。
  - ✓ <X> <Y>共に-200~200の間で指定する必要があります。
- ◇ modelscale (モデルスケール設定コマンド)
  - modelscale <scale> によりモデルのスケール(%)を指定する場合に使用します。
  - ✓ <scale>は1~1000の間で指定する必要があります。
- ◇ light (光源情報設定コマンド)
  - light <light-id> <X> <Y> <Z> <light-intensity> <ambient-intensity> により光源情報を設定する場合に使用します。
  - ✓ <light-id>は無効または有効のいずれかを設定します。
  - ✓ <X> <Y> <Z>は-32768~32767の間で指定する必要があります。
  - ✓ <light-intensity> <ambient-intensity>は0~4096の間で指定する必要があります。
  - ✓ <light-id>が無効の場合、<X> <Y> <Z> <light-intensity> <ambient-intensity>を省略可能です。
- ◇ fix (定型移動データ情報設定コマンド)
  - fix <fixedmove-index> <fit-mode> <move-mode> により定型移動時の動作を設定する場合に使用します。
  - ✓ <fixedmove-index>は定型移動データファイル内のどの移動ルートを使用するかを指定します。
  - ✓ <fit-mode>はフィットモードを指定します。

## &lt;fit-mode&gt;設定値

フィットモード	ストレッチ (デフォルト)	フィット (短辺長)	フィット (長辺長)	調整なし
設定値	0	1	2	3

- ✓ <move-mode>は定型移動モードを指定します。

## &lt;move-mode&gt;設定値

定型移動モード	絶対座標	相対 ノンエフェクト	相対 エリアループ	相対 エリアエンド
設定値	0	1	2	3

## ◇ detour (迂回情報設定コマンド)

detour <on/off> により移動禁止エリアが設定されていた場合に移動禁止エリアを迂回するか否かを設定する場合に使用します。

- ✓ <on/off>は無効または有効のいずれかを指定します。
- ✓ 迂回情報設定コマンドが省略された場合は、迂回有効として動作します。

## ◇ turn (旋回情報設定コマンド)

turn <turn> により自動移動実行時の旋回移動を設定する場合に使用します。

- ✓ <turn>は-5～5の間で指定する必要があります。
- ✓ 旋回情報設定コマンドが省略された場合は、旋回移動を行いません。

## ◇ speed (スピード設定コマンド)

speed <speed> により移動スピードを設定する場合に使用します。

- ✓ <speed>は0～10の間で指定する必要があります。
- ✓ <speed>が0に設定された場合は移動を行いません。
- ✓ スピード設定コマンドが省略された場合は、スピードを0として動作します。

## ◇ areaseize (マスコットエリアサイズ設定コマンド)

areaseize <size> <time> によりマスコットエリアサイズを設定する場合に使用します。

- ✓ <size>はマスコットエリア<sup>5</sup>サイズの種類を指定します。

## &lt;size&gt;設定値

マスコット エリアサイズ	Mサイズ	Sサイズ	フルサイ ズ ( 拡 大)	フルサイ ズ (シー ムレス) (拡大)	Lサイズ	LL サイズ
設定値	0	1	2	3	4	5

- ✓ <time>は1000～5000の間で指定する必要があります。単位はミリ秒です。
- ✓ <size>に0(標準 M サイズ)、1(縮小 S サイズ)、4(拡大Lサイズ)および5(拡大LLサイズ)を指定した場合は<time>に0を設定します。
- ✓ マスコットエリアサイズ設定コマンドが省略された場合は、M サイズとして動作します。

<sup>5</sup> マチキャラでキャラクタを表示するエリア。

ターゲットVRAMエリアの大きさとマスコットエリアサイズ設定コマンドの設定により、マスコットエリアサイズは以下のようになります。

ターゲットVRAMエリア		QCIF サイズ	QVGA サイズ	VGA サイズ	XGA サイズ
短辺長		96～239pixel	240～479pixel	480～719pixel	720～1280 pixel
マスコットエリア	Sサイズ	32 × 32pixel	48 × 48 pixel	96 × 96 pixel	144 × 144 pixel
	Mサイズ	48 × 48 pixel	64 × 64 pixel	128 × 128pixel	192 × 192pixel
	Lサイズ	64 × 64 pixel	96 × 96 pixel	192 × 192pixel	288 × 288pixel
	LLサイズ	96 × 96 pixel	128 × 128pixel	256 × 256pixel	384 × 384pixel

◇ conceal (隠しマスコット設定コマンド)

conceal <on/off> <action-file> <balloon> により隠しマスコットを設定する場合に使用します。

- ✓ <on/off>は無効または有効のいずれかを設定します。
- ✓ <on/off>が無効の場合は<action-file>および<balloon>を省略します。
- ✓ <action-file>は実行するアクションファイル名を指定します。
- ✓ <balloon>はバルーン情報を設定します。書式は“on/off QCIFbal-name QVGAbal-name VGAbal-name XGAbal-name”です。
- ✓ <balloon>の on/off は0(無効)または1(有効)のいずれかを指定します。
- ✓ <balloon>の QCIFbal-name はターゲットVRAMエリアが QCIF サイズの場合に使用するバルーン名を指定します。
- ✓ <balloon>の QVGAbal-name はターゲットVRAMエリアが QVGA サイズの場合に使用するバルーン名を指定します。
- ✓ <balloon>の VGAbal-name はターゲットVRAMエリアが VGA サイズの場合に使用するバルーン名を指定します。
- ✓ <balloon>の XGAbal-name はターゲットVRAMエリアが XGA サイズの場合に使用するバルーン名を指定します。
- ✓ <balloon>の QCIFbal-name、QVGAbal-name、VGAbal-name、XGAbal-name および XGAbal-name は存在するバルーン名である必要があります。
- ✓ <balloon>の on/off が無効の場合は、QCIFbal-name、QVGAbal-name、VGAbal-name、XGAbal-name および XGAbal-name は省略します。
- ✓ <balloon>の on/off が有効の場合は、QCIFbal-name、QVGAbal-name、VGAbal-name、XGAbal-name および XGAbal-name を省略することはできません。
- ✓ <balloon>の on/off が有効で該当サイズのバルーン画像が存在しない場合は、バルーン非表示とし、該当する項目を-1に設定します。
- ✓ <balloon>はバルーン設定が条件なしの場合は省略します。

◇ balloon (バルーン情報設定コマンド)

balloon <on/off> <QCIFbal-name> <QVGAbal-name> <VGAbal-name> <XGAbal-name>により、バルーン情報を設定する場合に使用します。

- ✓ <on/off>は0(無効)または1(有効)のいずれかを指定します。
- ✓ <QCIFbal-name>はターゲットVRAMエリアが QCIF サイズの場合に表示するバルーン名を設定します。
- ✓ <QVGAbal-name>はターゲットVRAMエリアが QVGA サイズの場合に表示するバルーン名を設定します。
- ✓ <VGAbal-name>はターゲットVRAMエリアが VGA サイズの場合に表示するバルーン名を設定します。
- ✓ <XGAbal-name>はターゲットVRAMエリアが XGA サイズの場合に表示するバルーン名を設定します。
- ✓ <QCIFbal-name>、<QVGAbal-name>、<VGAbal-name>および<XGAbal-name>は存在するバルーン名を指定する必要があります。
- ✓ バルーン設定が条件なしの場合はバルーン情報を省略します。
- ✓ <on/off>が無効の場合は、<QCIFbal-name>、<QVGAbal-name>、<VGAbal-name>および<XGAbal-name>は省略します。
- ✓ <on/off>が有効の場合は、<QCIFbal-name>、<QVGAbal-name>、<VGAbal-name>および<XGAbal-name>を省略することはできません。
- ✓ <on/off>が有効で該当サイズのバルーン画像が存在しない場合は、バルーン非表示とし、該当する項目に-1を設定します。

◇ useful (使用期間設定コマンド)

useful <id> <value> によりイベント使用期間を設定する場合に使用します。

- ✓ <id>は使用期間の指定方法を指定します。
- ✓ <id>に設定できる値は、0(設定なし)、1(使用禁止期間)および2(使用解除年月



日時間)です。

- ✓ <value>はイベントの使用禁止期間または使用解除年月日時を指定します。
- ✓ <id>に0が設定された場合は、<value>に0を設定します。

◇ ratio (実行比率設定コマンド)

ratio <ratio> によりイベント実行比率を設定する場合に使用します。

- ✓ <ratio>はイベントの実行比率を設定します。

◇ scene (重畳シーン別イベント制御設定コマンド)

scene <on/off> <scene-no> により重畳シーン別イベント制御を設定する場合に使用します。

- ✓ <on/off>は無効または有効のいずれかを指定します。
- ✓ <scene-no>は重畳シーンを00～80の間で指定する必要があります。

<scene-no>設定値

重畳 シーン	待ち受 け	ブラウ ザ	メーラ	テレビ 電話	メニ ュ ー	メール 受信確 認	i コン シ ェ ル TOP	音声入 力モー ド	その他
設定値	10	20	30	40	50	60	70	80	00

◇ unread (未読メール数設定コマンド)

unread <unreadmail>により、イベントを有効とする未読メール数を設定する場合に使用します。

- ✓ <unreadmail>はイベントを有効とする未読メール数を指定します。
- ✓ <unreadmail>は0～65535の範囲で指定する必要があります。

◇ call (通話時間設定コマンド)

call <time>により、イベントを有効とする通話時間を設定する場合に使用します。

- ✓ <time>はイベントを有効とする通話時間を秒単位で設定します。
- ✓ <time>は0～35999999の範囲で指定する必要があります。

◇ play (目的状態設定コマンド)

play <motor-file:state> により目的状態の設定 'を行う場合に使用します。

- ✓ 指定する motor-file は事前に motor セクションにて定義されていなければなりません。
- ✓ 指定する state は motor-file にて定義されている state でなければなりません。
- ✓ motor-file と state の間は:(コロン)で区切ります。
- ✓ 複数指定を行う場合は各状態を空白で区切ります。  
例) play { motorA:stateA motorB:stateB motorC:stateC }
- ✓ 同一モータに対する複数指定はできません。  
例) play { motorA:stateA motorA:stateB } ... ×  
同一モータ(motorA)に対する複数指定なので不可
- ✓ motor-file および state は[特殊文字](#)及び[トークン](#)は使用できません。

◇ set (現在状態設定コマンド)

set <motor-file:state> により現在の状態の設定 'を行う場合に使用します。

- ✓ 指定する motor-file は事前に motor セクションにて定義されていなければなりません。
- ✓ 指定する state は motor-file にて定義されている state でなければなりません。
- ✓ motor-file と state の間は:(コロン)で区切ります。
- ✓ 複数指定を行う場合は各状態を空白で区切ります。  
例) set { motorA:stateA motorB:stateB motorC:stateC }
- ✓ 同一モータに対する複数指定はできません。  
例) set { motorA:stateA motorA:stateB } ... ×  
同一モータ(motorA)に対する複数指定なので不可
- ✓ motor-file および state は[特殊文字](#)及び[トークン](#)は使用できません。

### S.3.1.10. ビヘイビアファイル設定例

ビヘイビアファイルの設定例を以下に示します。

```
! Sample mascot behavior
behavior {
    model          { sample.bac }                ... ①
    texture        { texture1.bmp texture2.bmp } ... ②
    map            { map1.bmp }                  ... ③
    motor          { feeling.mot head.mot body1.mot } ... ④
    move           { move.fmf }                 ... ⑤
    balloon        { balloon.bln }              ... ⑥
    dress          {                             ... ⑦
        20110000 1 0 0800-1800 dress1.tra 1 bal1 -1 bal2 -1;
        default dress2.tra;
    }
    default {
        D {
            F00 {                                ... ⑧
                setbg wait.bmp;
                light 1 100 0 0 4096 2048;
                set { feeling.mot:wait head.mot: wait body1.mot: wait }
            }
        }
        W05 {
            F00 {                                ... ⑨
                set { feeling.mot:1_feeling head.mot:1_head body1.mot:1_body }
            }
        }
        X07 {
            F00 {                                ... ⑩
                set { feeling.mot:2_feeling head.mot:2_head body1.mot:2_body }
            }
        }
        Y12 {
            F00 {                                ... ⑪
                set { feeling.mot:3_feeling head.mot:3_head body1.mot:3_body }
            }
        }
        A00 {
            S00 {                                ... ⑫
                setbg glad.bmp;
                set { feeling.mot:4_feeling head.mot:4_head body1.bot:4_body }
            }
            F00 {                                ... ⑬
                set { feeling.mot:5_feeling head.mot:5_head body1.bot:5_body }
            }
            M00 {                                ... ⑭
                set { feeling.mot:6_feeling head.mot:6_head body1.bot:6_body }
            }
        }
    }
}
```

```

Z18 {
  F00 {
    ... ⑮
    set { feeling.mot:7_feeling head.mot:7_head body1.mot:7_body }
  }
}
P {
  F00 {
    ... ⑯
    balloon 0;
    set { feeling.mot:8_feeling head.mot:8_head body1.mot:8_body }
  }
}
N {
  F00 {
    ... ⑰
    unread 10;
    balloon 1 bal3 bal4 bal5 bal6;
    set { feeling.mot:9_feeling head.mot:9_head body1.mot:9_body }
  }
}
M {
  F00 {
    ... ⑱
    balloon 0;
    set { feeling.mot:10_feeling head.mot:10_head body1.mot:10_body }
  }
}
C {
  F00 {
    ... ⑲
    call 600;
    balloon 1 -1 -1 bal7 -1;
    set { feeling.mot:11_feeling head.mot:11_head body1.mot:11_body };
  }
}
}
schedule20 {
  D {
    F00 {
    ... ⑳
    set { feeling.mot:12_feeling head.mot:12_head body1.mot:12_body }
    }
  }
}
}

```

- ① モデルファイルとして「sample.bac」を使用します。
- ② テクスチャファイルとして「texture1.bmp」、「texture2.bmp」を使用します。
- ③ 環境マッピングファイルとして「map1.bmp」を使用します。
- ④ モーターファイルとして「feeling.mot」、「head.mot」、「body1.mot」を使用します。
- ⑤ 定型移動データファイルとして「move.fmf」を使用します。
- ⑥ バルーンデータファイルとして「balloon.blm」を使用します。
- ⑦ 時間反映表現の情報を設定します。
  1. 2011年春(3月～5月)の8:00～18:00の間「dress1.tra」を使用するように設定する。
  2. 時間反映条件に当てはまらない期間は「dress2.tra」を使用するように設定す

- る。
3. 時間反映中はターゲットVRAMエリアが QCIF サイズの場合、バルーン画像に bal1 を使用するように設定する。
  4. 時間反映中はターゲットVRAMエリアが QVGA サイズの場合、バルーン画像を表示しないように設定する。
  5. 時間反映中はターゲットVRAMエリアが VGA サイズの場合、バルーン画像に bal2 を使用するように設定する。
  6. 時間反映中はターゲットVRAMエリアが XGA サイズの場合、バルーン画像を表示しないように設定する。
- ⑧ 通常マスコットデフォルトイベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 背景画像に「wait.bmp」を設定する。
  2. 光源情報を以下のように設定する
 

・<light-id>	1
・<X> <Y> <Z>	100 0 0(左からの光源)
・<light-intensity>	4096
・<ambient-intensity>	2048
  3. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの wait ステート
    - ・head.mot モータの wait ステート
    - ・body1.mot モータの wait ステート
- ⑨ 通常マスコットアラーム[05]イベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 1\_feeling ステート
    - ・head.mot モータの 1\_head ステート
    - ・body1.mot モータの 1\_body ステート
- ⑩ 通常マスコットダイアログ[07]イベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 2\_feeling ステート
    - ・head.mot モータの 2\_head ステート
    - ・body1.mot モータの 2\_body ステート
- ⑪ 通常マスコットiコンシェル TOP ショートカットフォーカス[12]イベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 3\_feeling ステート
    - ・head.mot モータの 3\_head ステート
    - ・body1.mot モータの 3\_body ステート
- ⑫ 通常マスコットiコンシェル[00]イベント状態定登場イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 背景画像に「glad.bmp」を設定する。
  2. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 4\_feeling ステート
    - ・head.mot モータの 4\_head ステート
    - ・body1.mot モータの 4\_body ステート
- ⑬ 通常マスコットiコンシェル[00]イベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 5\_feeling ステート

- ・head.mot モータの 5\_head ステート
  - ・body1.mot モータの 5\_body ステート
- ⑭ 通常マスコットiコンシェル[00]イベント状態移動イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 6\_feeling ステート
    - ・head.mot モータの 6\_head ステート
    - ・body1.mot モータの 6\_body ステート
- ⑮ 通常マスコットiコンシェル TOP[18]イベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 7\_feeling ステート
    - ・head.mot モータの 7\_head ステート
    - ・body1.mot モータの 7\_body ステート
- ⑯ 通常マスコット不在着信イベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. バルーン情報を非表示にする。
  2. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 8\_feeling ステート
    - ・head.mot モータの 8\_head ステート
    - ・body1.mot モータの 8\_body ステート
- ⑰ 通常マスコット未読メールありイベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 未読メールが10件以上ある場合にイベントが有効になるように設定する。
  2. バルーン情報を以下のように設定する。
    - ・ターゲット VRAM エリアが QCIF サイズの場合、バルーン画像に「bal3」を使用するように設定する。
    - ・ターゲット VRAM エリアが QVGA サイズの場合、バルーン画像に「bal4」を使用するように設定する。
    - ・ターゲット VRAM エリアが VGA サイズの場合、バルーン画像に「bal5」を使用するように設定する。
    - ・ターゲット VRAM エリアが XGA サイズの場合、バルーン画像に「bal6」を使用するように設定する。
  3. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 9\_feeling ステート
    - ・head.mot モータの 9\_head ステート
    - ・body1.mot モータの 9\_body ステート
- ⑱ 通常マスコットメール受信イベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. バルーン情報を非表示にする。
  2. 各パーツの状態を以下のように設定する。
    - ・feeling.mot モータの 10\_feeling ステート
    - ・head.mot モータの 10\_head ステート
    - ・body1.mot モータの 10\_body ステート
- ⑲ 通常マスコット通話時間イベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。
1. 通話時間が600秒異常の場合にイベントが有効になるように設定する。
  2. バルーン情報を以下のように設定する。
    - ・ターゲットVRAMエリアが QCIF サイズの場合、バルーン画像を非表示に設定する。

- ・ターゲットVRAMエリアが QVGA サイズの場合、バルーン画像を非表示に設定する。
- ・ターゲットVRAMエリアが VGA サイズの場合、バルーン画像に「bal7」を使用するように設定する。
- ・ターゲットVRAMエリアが XGA サイズの場合、バルーン画像を非表示に設定する。

3. 各パーツの状態を以下のように設定する。

- ・feeling.mot モータの 11\_feeling ステート
- ・head.mot モータの 11\_head ステート
- ・body1.mot モータの 11\_body ステート

②⑩ 時間指定(20)マスコットデフォルトイベント状態定位置イベント[00]が発生した場合の動作として以下のような動作を設定します。

1. 各パーツの状態を以下のように設定する。

- ・feeling.mot モータの 12\_feeling ステート
- ・head.mot モータの 12\_head ステート
- ・body1.mot モータの 12\_body ステート

### S.3.2. モータファイル

モータは以下に説明するステートの定義によって、マチキャラの状態の遷移を行います。  
 テキストファイル(拡張子は.mot)として作成し、モータ毎に別のファイルとして作成します。  
 モータファイル名は**特殊文字**を含めることはできません。  
 モータファイルの漢字コードはシフトJISで作成します。

ここではモータファイル内での定義について説明します。  
 モータを定義するための構文は以下のように定義されます。

#### 3D コンテンツ

```
<state-name> : <action-file> {
    [ [ start { <command> [ <command>... ] ] ]
    [ end { <command> [ <command>... ] ] ]
    [ time <msec> { <command> [ <command>... ] ] ]
    [ on <handler> { <command> [ <command>... ] ] ];
}
```

#### 2D コンテンツ

```
<state-name> : <gif1-file> <gif2-file> <gif3-file> <gif4-file> <gif5-file>
               <gif6-file> <gif7-file> <gif8-file> {
    [ [ start { <command> [ <command>... ] ] ]
    [ end { <command> [ <command>... ] ] ]
    [ on <handler> { <command> [ <command>... ] ] ];
}[ <play-time> ];
```

- ◇ <state-name>(ステート名)  
 ステート名を定義する文字列です。
  - ✓ **特殊文字**、**トークン**を含まない文字列で16バイト以内で指定します。
  - ✓ 定義可能なステート数の最大は256です。
- ◇ <action-file>(アクションファイル名)  
 ステート名に続き、:(コロン)の次に実行するアクションファイルを指定します。
  - ✓ 漢字ファイル名は使用可能ですが、ファイル名に**特殊文字**は使用できません。
  - ✓ アクションファイル名に**特殊文字**を除く**トークン**を含める場合は二重引用符で囲む必要があります。
  - ✓ 複数のアクションファイルは定義できません。
  - ✓ アクションファイル名は1ステートに1つ定義可能です。
- ◇ <gif1-file>...<gif8-file>(GIF ファイル名)  
 ステート名に続き、:(コロン)の次に実行するGIFファイルを指定します。
  - ✓ 差分アニメーションの GIF ファイルは正常に表示できない場合があります。
  - ✓ 漢字ファイル名は使用可能ですが、ファイル名に**特殊文字**は使用できません。
  - ✓ アクションファイル名に**特殊文字**を除く**トークン**を含める場合は二重引用符で囲む必要があります。
  - ✓ 移動以外の場合は全て同一の GIF ファイルを指定します。
  - ✓ 正面方向から時計回りに GIF ファイルを指定します。複数方向で同じ GIF ファイルを使用する場合は、同一ファイル名を該当箇所に記述します。

- ◇ <play-time>(最大アニメーション時間)  
最大アニメーション時間を指定します。
  - ✓ <playtime>は1～10000の間で指定する必要があります。単位はミリ秒です。
  - ✓ 最大アニメーション時間を指定しない場合は1000ミリ秒とします。
- ◇ start / end / time / on (イベント)  
ステート内で処理するイベントを設定します。イベントについてはイベント(motor)で説明します。
  - ✓ イベントは1ステート内で重複して定義することはできません。



### S.3.2.1. イベント (motor)

- ◇ start (開始イベント)  
このイベントを定義すると、定義されているステートのアクションの前にこのイベント内のコマンドが実行されます。
  - ◇ end (終了イベント)  
このイベントを定義すると、定義されているステートのアクションの後にこのイベント内のコマンドが実行されます。
  - ◇ time <msec> (間隔イベント)  
このイベントを定義すると、定義されたステートに遷移してから指定された秒数 (msec) 後に、このイベント内のコマンドが実行されます。
  - ◇ on <state> または <\*> (イベント)  
このイベントを定義すると、定義されたステートに遷移してから指定されたステート (目的状態の設定) があればこのイベント内のコマンドが実行されます。  
\* (アスタリスク) を指定した場合は、どのステートが目的状態として設定されてもイベント内のコマンドが実行されます。
- ✓ ステートを定義する場合はモータ内に存在するステートである必要があります。

### S.3.2.2.コマンド詳細(motor)

各イベントで実行できるコマンドについて説明します。

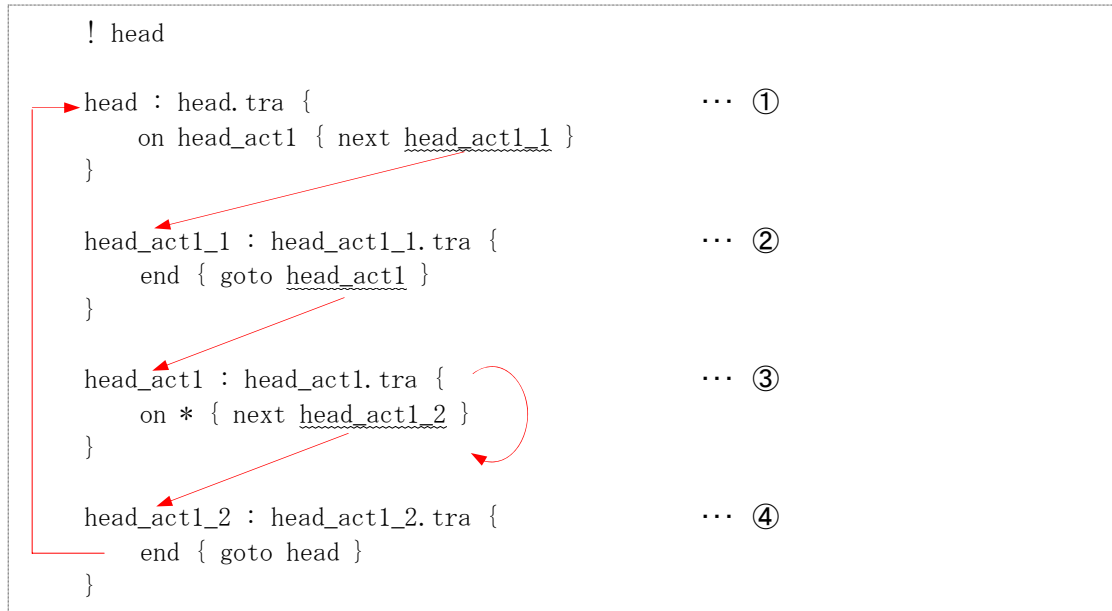
- ✓ 複数のコマンドを指定する場合は行末に; (セミコロン)を設定する必要があります。
- ◇ setfg (前景設定コマンド) / setbg (背景設定コマンド)
  - setfg <foreground-file> [attribute]により前景画像を指定する場合に使用します。
  - setbg <background-file> [attribute]により背景画像を指定する場合に使用します。
  - ✓ 存在するファイルである必要があります。
  - ✓ 指定できる画像ファイルは GIF89a または Bitmap 形式です。
  - ✓ ファイル名に漢字は使用できませんが、[特殊文字](#)は使用できません。
  - ✓ 前景／背景ファイル名に[特殊文字](#)を除く[トークン](#)を含める場合は二重引用符で囲む必要があります。
  - ✓ setfg/setbg コマンドに対し、前景／背景画像の設定は必須です。
  - ✓ [attribute]には属性として以下の内容が設定可能です。

tile	タイリング
zoom	ストレッチ
center	センタリング

何も指定しない場合は「なし」となります。
  - ✓ 設定されている画像について同じ属性が指定されている必要があります。
  - ✓ マチキャラでは、前景／背景として設定する画像ファイルの属性は全て「zoom」として扱います。
- ◇ next (遷移先設定コマンド)
  - next <state> により遷移する状態を事前に設定しておくことができます。
  - ✓ 同一モータ内に存在するステートを指定する必要があります。
  - ✓ 遷移先設定コマンドでは内部の現在状態は変更されません。ステータスは gotonext コマンドにより遷移します。
  - ✓ gotonext コマンドが設定されていない状態でコマンドが終了した場合は、遷移先設定コマンドで指定されたステートに遷移します。
- ◇ goto (遷移コマンド)
  - goto <state> により状態を遷移する先のステートを指定します。内部の動作としては現在状態を指定された遷移先に変更します。
  - ✓ 同一モータ内に存在するステートを指定する必要があります。
- ◇ gotonext (遷移コマンド)
  - gotonext により next コマンドで指定されたステートへの遷移が行われます。

### S.3.2.3. モータファイル設定例

モータファイルの設定例を以下に示します。  
3D コンテンツ



#### ①head ステート

head ステートに遷移すると以下の動作を行います。

1. 「head.tra」のアクションを実する。
2. 目的状態として head\_act1 の設定が行われていない場合は、「head.tra」が繰り返される。
3. 目的状態が on head\_act1 に設定されると、head\_act1 イベント指定によりコマンド next head\_act1\_1 が実行され、次に実行すべきステートが設定される。
4. コマンドが終了すると自動的に end イベントが発行され、現在状態のステートは head\_act1\_1 に遷移します。

#### ②head\_act1\_1 ステート

head\_act1\_1 ステートに遷移すると以下の動作を行います。

1. 「head\_act1\_1.tra」のアクションを実行する。
2. 実行後、goto head\_act1 により head\_act1 ステートに遷移する。

#### ③head\_act1 ステート

head\_act1 ステートに遷移すると以下の動作を行います。

1. ①のイベント指定で指定された目的状態と現在の状態が一致したので、目的状態への遷移は終了する。
2. 「head\_act1.tra」が実行される。
3. ステート内に on \* イベント指定があるので、目的状態の変更があると、次のコマンドが実行される。
4. next head\_act1\_2 により次に実行すべきステートが設定される。  
ここでのポイントは next コマンドにより現在状態を変更することなく状態を遷移させることです。

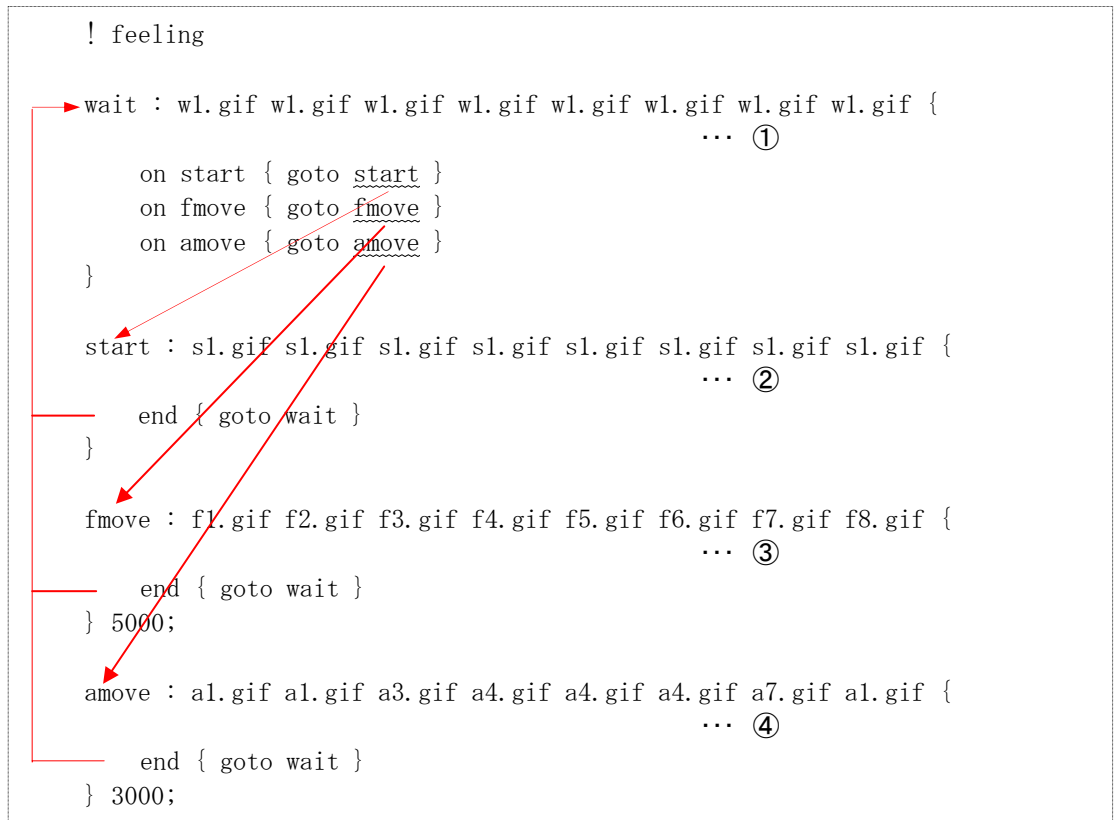
#### ④head\_act1\_2 ステート

head\_act1\_2 ステートに遷移すると以下の動作を行います。

1. 「head\_act1\_2.tra」のアクションを実行する。
2. 実行後、end イベントにより goto head コマンドが実行され、head ステートへ遷移する。

3. head ステートへ遷移後は特に目的状態が設定されていないので、「head.tra」のアクションが繰り返される。

## 2D コンテンツ



### ①wait ステート

wait ステートに遷移すると以下の動作を行います。

1. 「w1.gif」のアクションが実行される。
2. on イベントで指定されたステートに目的状態が変更されると次のコマンドが実行される。
3. 目的状態として start/fmove/amove の設定が行われていない場合には「w1.gif」のアクションが繰り返される。
4. 目的状態が start に設定されると、start イベント指定によりコマンド goto start が実行され、次に実行すべきステートが設定される。
5. コマンドが終了すると自動的に end イベントが発生し現在状態のステートは start ステートに遷移する。fmove/amove に設定された場合も同様に goto fmove/goto amove が実行され fmove ステート/amove ステートに遷移する。

### ②start ステート

start ステートに遷移すると以下の動作を行います。

1. 「s1.gif」のアクションを実行する。
2. 最大アニメーション時間が指定されていないので1000ミリ秒経過後、goto wait により、wait ステートに遷移する。

### ③fmove ステート

fmove ステートに遷移すると以下の動作を行います。

1. 「f1.gif」「f2.gif」「f3.gif」「f4.gif」「f5.gif」「f6.gif」「f7.gif」「f8.gif」のアクションを実行する。どのアクションが実行されるかは、キャラクタの進行方向によって決まる。
2. 5000ミリ秒後に end イベントが発生し、goto wait により wait ステートに遷移する。

る。

④amove ステート

amove ステートに遷移すると以下の動作を行います。

1. 「a1.gif」「a3.gif」「a4.gif」「a7.gif」のアクションを実行する。どのアクションが実行されるかは、キャラクタの進行方向によって決まる。
2. 3000ミリ秒後に end イベントが発生し、goto wait により wait ステートに遷移する。

### S.3.3. マチキャラメイクファイル

マチキャラメイクファイルは以降に説明する各セクション毎の定義によって、マスコット毎の状態の設定を行います。テキストファイルとして作成します(拡張子は.cmf)。マチキャラメイクファイル名に**特殊文字**を含めることはできません。マチキャラメイクファイルの漢字コードはシフトJISで作成します。

ここではマチキャラメイクファイル内でのセクション定義について説明します。マスコットを定義するための構文は以下のように定義されます。

```
<mascot-name> {
  version 5.00;
  distr <id>;
  id <mascot-id>;
  title <mascot-title>;
  [ url <url>; ]
  popup <popup-file>;
  [ contentid <content-id>; ]
  [ fileid <file-id>; ]
  [ mmdurladdress <mmd-address>; ]
  [ mmdurlquery <mmd-query>; ]
  mascot {
    behavior ビヘイビアスクリプトファイル名;
    [ useful 0 0 0 0 0 0; ]
    [ preview プレビューデータファイル名; ]
    [ subtitle マスコットサブタイトル; ]
    [ cg 2D/3D フラグ; ]
    [ mask_col 透過色データ; ]
    [ pers angle near-clip far-clip; ]
    [ view up-x up-y up-z pos-x pos-y pos-z look-x look-y look-z; ]
    [ light id x y z light ambient; ]
  }
  [ mascot {
    behavior ビヘイビアスクリプトファイル名;
    [ useful 0 0 0 0 0 0; ]
    [ preview プレビューデータファイル名; ]
    [ subtitle マスコットサブタイトル; ]
    [ cg 2D/3D フラグ; ]
    [ mask_col 透過色データ; ]
    [ pers angle near-clip far-clip; ]
    [ view up-x up-y up-z pos-x pos-y pos-z look-x look-y look-z; ]
    [ light id x y z light ambient; ]
  } ... ]
  [ cfdurl <cfdurl>; ]
  [ cfdtime <flag> <time>; ]
  filename <cfd-file>;
}
```

☆ <mascot-name>(マチキャラ名)

マチキャラ名を定義する文字列です。

- ✓ マチキャラメイクファイルには1つのマチキャラのみ定義できます。
- ✓ **特殊文字**、**トークン**は使用できません。

- ◇ version distr id title url popup cfdurl cfdtime contentid fileid mmdurladdress mmdurlquery mascot (セクション)  
各セクションには定義する順番に指定があります。  
以下の点に考慮して定義してください。他はどの順番でも構いません。
  - ✓ popup セクションは mascot セクションより前に定義すること。
- ◇ 重複定義  
mascot を除く各セクションは重複して定義できません。

### S.3.3.1. version セクション

version セクションでは、マチキャラを使用するアバタエンジンのバージョンを指定します。  
アラーム、ダイアログ、iコンシェル TOP ショートカットフォーカス、iコンシェル TOP イベントおよび  
ネットワークアップデート機能対応マチキャラでは「5.00」を指定します。

version セクションは以下のように定義されます。

```
version 5.00
```

◇ version セクション

✓ version セクションの複数定義はできません。



### S.3.3.2. distr セクション

distr セクションでは、マチキャラデータを再配布可能にするかどうか指定します。

distr セクションは以下のように定義されます。

```
distr <id>
```

☆ distr セクション

✓ distr セクションの複数定義はできません。

☆ <id> (再配布識別子)

マチキャラデータを再配布可能にするかどうかを指定します。

✓ 0(再配布不可)または1(再配布可)を設定する必要があります。

例) 再配布不可に設定する場合。

```
distr 0
```

### S.3.3.3. id セクション

id セクションでは、マチキャラデータの ID を定義します。

id セクションは以下のように定義されます。

```
id <mascot-id>
```

☆ id セクション

- ✓ id セクションの複数定義はできません。
- ✓ id セクションを省略することはできません。

☆ <mascot-id> (マスコット ID)

マスコット ID を指定します。

- ✓ 定義できるマスコット ID の数は1個です。
- ✓ マスコット ID には半角の A～Z、a～z および 0～9 を使用できます。
- ✓ マスコット ID を””で囲みます。

例) マスコット ID に「SAMPLE」と設定する場合。

```
id “SAMPLE”
```

#### S.3.3.4. title セクション

title セクションでは、マスコットデータのタイトルを指定します。

title セクションは以下のように定義されます。

```
title <mascot-title>
```

- ◇ title セクション
  - ✓ title セクションの複数定義はできません。
  - ✓ title セクションを省略することはできません。
- ◇ <mascot-title> (マスコットタイトル)
  - ✓ 定義できるマスコットタイトルの数は1個です。
  - ✓ マスコットタイトルを””で囲みます。
  - ✓ マスコットタイトルに[特殊文字](#)は使用できません。

例) マスコットタイトルとして「mascot title」を使用する場合。

```
title “mascot title”
```

### S.3.3.5. url セクション

url セクションでは、データにリンクさせる URL を指定します。

url セクションは以下のように定義されます。

```
url <url>
```

- ◇ url セクション
  - ✓ url セクションの複数定義はできません。
- ◇ <url> (更新先 URL)
  - ✓ 存在する URL である必要があります。
  - ✓ 定義できる更新先 URL 数は最大1個です。
  - ✓ 更新先 URL を””で囲みます。

例) 更新先 URL として「http://www.sample.co.jp」を使用する場合。

```
url “http://www.sample.co.jp”
```

### S.3.3.6. popup セクション

popup セクションでは、作成するポップアップデータファイルの名称を設定します。

popup セクションは以下のように定義されます。

```
popup <popup-file>
```

◇ popup セクション

- ✓ popup セクションの複数定義はできません。
- ✓ i コンシェル機能対応のマチキャラを作成する場合、popup セクションを省略することはできません。

◇ <popup-file> (ポップアップデータファイル名)

ポップアップデータファイル名(.pop)を指定します。

- ✓ 定義できるポップアップデータファイル数は1個です。
- ✓ <popup-file>は存在するファイルである必要があります。
- ✓ 漢字ファイル名は使用できますが、ファイル名に[特殊文字](#)は使用できません。
- ✓ ポップアップデータファイル名に[特殊文字](#)を除く[トークン](#)を含める場合は二重引用符で囲む必要があります。

例) ポップアップデータファイル名として「sample.pop」を使用する場合。

```
popup sample.pop
```

### S.3.3.7. contentid セクション

contentid セクションでは、マチキャラデータの CFD コンテンツ ID を定義します。

contentid セクションは以下のように定義されます。

```
contentid <content-id>
```

✧ contentid セクション

- ✓ contentid セクションの複数定義はできません。

✧ <content-id> (コンテンツ ID)

コンテンツ ID を指定します。

- ✓ 定義できるコンテンツ ID の数は1個です。
- ✓ コンテンツ ID は 00000000000000000000000000000000 です。
- ✓ コンテンツ ID を""で囲みます。

例) コンテンツ ID に「00000000000000000000000000000000」と設定する場合。

```
contentid "00000000000000000000000000000000"
```

### S.3.3.8. fileid セクション

fileid セクションでは、マチキャラデータの CFD ファイル ID を定義します。

fileid セクションは以下のように定義されます。

```
fileid <file-id>
```

☆ fileid セクション

✓ fileid セクションの複数定義はできません。

☆ <file-id> (ファイル ID)

ファイル ID を指定します。

✓ 定義できるファイル ID の数は1個です。

✓ ファイル ID には半角の A～Z、a～z および 0～9 を使用できます。

✓ ファイル ID を””で囲みます。

例) ファイル ID に「3DSampleContentFID30000100000001」と設定する場合。

```
fileid “3DSampleContentFID30000100000001”
```

### S.3.3.9. mmdurladdress セクション

mmdurladdress セクションでは、MMD ファイルの取得先 URL(アドレス部)を定義します。

mmdurladdressid セクションは以下のように定義されます。

```
mmdurladdress <mmd-address>
```

☆ mmdurladdress セクション

- ✓ mmdurladdress セクションの複数定義はできません。

☆ <mmd-address> (MMD ファイル取得先 URL アドレス部)

MMD ファイル取得先 URL のアドレス部を指定します。

- ✓ 存在する URL である必要があります。
- ✓ 定義できる MMD ファイル取得先 URL アドレス部の数は最大1個です。
- ✓ MMD ファイル取得先 URL アドレス部を””で囲みます。

例) MMD ファイル取得先 URL アドレス部に「http://www.sample.co.jp/contents」と設定する場合。

```
mmdurladdress “http://www.sample.co.jp/contents”
```



### S.3.3.10. mmdurlquery セクション

mmdurlquery セクションでは、MMD ファイルの取得先 URL(クエリ部)を定義します。

mmdurlquery セクションは以下のように定義されます。

```
mmdurlquery <mmd-query>
```

- ☆ mmdurlquery セクション
  - ✓ mmdurlquery セクションの複数定義はできません。
- ☆ <mmd-query> (MMD ファイル取得先 URL クエリ部)  
MMD ファイル取得先 URL のクエリ部を指定します。
  - ✓ 存在する URL である必要があります。
  - ✓ 定義できる MMD ファイル取得先 UR クエリの数は最大1個です。
  - ✓ MMD ファイル取得先 URL クエリ部を””で囲みます。

例) MMD ファイル取得先 URL クエリ部に「content?type=1&id=1」と設定する場合。

```
mmdurlquery “content?type=1&id=1”
```

### S.3.3.11. mascot セクション

mascot セクションでは、マスコットの状態を設定します。

mascot セクションは以下のように定義されます。

```
mascot {
  behavior <behavior-file>;
  [ useful <flag> <date> <time> <talktime> <rcvmail> <sndmail>; ]
  [ preview <preview-file>; ]
  [ subtitle <mascot-subtitle>; ]
  [ cg <flag>; ]
  [ mask_col <color>; ]
  [ pers <angle> <near-clip> <far-clip>; ]
  [ view <up-x> <up-y> <up-z>
        <pos-x> <pos-y> <pos-z> <look-x> <look-y> <look-z>; ]
  [ light <id> [ <x> <y> <z> <light> <ambient> ]; ]
}
```

#### ◇ mascot セクション

- ✓ mascot セクションは最低1個の設定が必要です。
- ✓ poup セクションより後に定義する必要があります。

以下にマスコットセクションで設定する項目について説明します。

#### ◇ behavior (ビヘイビア)

behavior <behavior-file>によりマスコットで使用するビヘイビアを定義します。

- ✓ ビヘイビアの複数定義はできません。
- ✓ ビヘイビアを省略することはできません。
- ✓ <behavior-file>は存在するファイルである必要があります。
- ✓ <behavior-file>は漢字は使用できませんが、[特殊文字](#)は使用できません。
- ✓ <behavior-file>に[特殊文字](#)を除く[トークン](#)を含める場合は二重引用符で囲む必要があります。

#### ◇ useful (マスコット使用期間)

useful <flag> <date> <time> <talktime> <rcvmail> <sndmail>によりマスコットの使用条件を定義します。

- ✓ useful コマンドの複数定義はできません。
- ✓ <flag>は使用条件フラグを設定します。
- ✓ <flag>に設定できる値は、0(設定なし)、1(AND条件)および2(OR条件)です。
- ✓ <date>は使用解除年月日時間を設定します。
- ✓ <time>は使用禁止期間を設定します。
- ✓ <talktime>は積算通話時間を設定します。
- ✓ <talktime>は0～35999999の間で設定する必要があります。
- ✓ <rcvmail>はメール受信数を設定します。
- ✓ <rcvmail>は0～65535の間で設定する必要があります。
- ✓ <sndmail>はメール送信数を設定します。
- ✓ <sndmail>は0～65535の間で設定する必要があります。
- ✓ 指定しない項目は0を設定します。

- ✓ <flag>に0を設定した場合は、<date><time><talktime><rcvmail><sndmail>は全て0を設定します。
  - ✓ <flag>に0を設定した場合は、該当マスコットは常に使用可能になります。
  - ✓ <flag>に1を設定した場合は、<date><time><talktime><rcvmail><sndmail>で指定した条件を全て満たした場合に該当マスコットが使用可能になります。
  - ✓ <flag>に2を設定した場合は、<date><time><talktime><rcvmail><sndmail>で指定した値のいずれかを満たした場合に該当マスコットが使用可能になります。
- ◇ preview (プレビューイメージ)
- preview <preview-file>によりマスコットで使用するプレビューイメージを定義します。
- ✓ プレビューイメージの複数定義はできません。
  - ✓ <preview-file>は存在するファイル名である必要があります。
  - ✓ <preview-file>は漢字は使用できますが、[特殊文字](#)は使用できません。
  - ✓ <preview-file>に[特殊文字](#)を除く[トークン](#)を含める場合は二重引用符で囲む必要があります。
- ◇ subtitle (マスコットサブタイトル)
- subtitle <mascot-subtitle>によりマスコットサブタイトルを定義します。
- ✓ マスコットサブタイトルの複数定義はできません。
  - ✓ マスコットサブタイトルを""で囲みます。
  - ✓ <mascot-subtitle>は漢字は使用可能ですが、[特殊文字](#)は使用できません。
- ◇ cg (2D/3D フラグ)
- cg <flag>によりマスコットが 2D または 3D かを指定します。
- ✓ 2D/3D フラグの複数定義はできません。
  - ✓ <flag>は0(3D)または1(2D)を設定します。
- ◇ mask\_col (透過色)
- mask\_col <color>により背景の透過色を定義します。
- ✓ 透過色の複数定義はできません。
  - ✓ 透過色はRGB888形式で指定します。
  - ✓ <color>は赤、緑、青をそれぞれ0~255の間で指定する必要があります。
- ◇ pers (透視投影情報)
- pers <angle> <near-clip> <far-clip>により透視投影情報を定義します。
- ✓ 透視投影情報の複数定義はできません。
  - ✓ 透視投影情報の定義を省略した場合、平行投影となります。
  - ✓ <angle>は 1~179の間で指定する必要があります。
  - ✓ <near-clip> < <far-clip>である必要があります。
- ◇ view (視野情報)
- view <up-x> <up-y> <up-z> <pos-x> <pos-y> <pos-z> <look-x> <look-y> <look-z>により視野情報を定義します。
- ✓ 視野情報の複数定義はできません。
  - ✓ 各項目は-32768~32767の間で指定する必要があります。
- ◇ light (平行光源情報)
- light <id> <x> <y> <z> <light> <ambient>により平行光源情報を定義します。
- ✓ 平行光源情報の複数定義はできません。
  - ✓ <id>は無効または有効のいずれかを設定します。

- ✓ <X> <Y> <Z>は-32768～32767の間で指定する必要があります。
- ✓ <light> <ambient>は0～4096の間で指定する必要があります。
- ✓ <id>が無効の場合、<X> <Y> <Z> <light> <ambient>は省略可能です。

### S.3.3.12. cfdurl セクション

cfdurl セクションでは、マチキャラメタデータファイルに記載するマチキャラデータのダウンロード URL を指定します。

cfdurl セクションは以下のように定義されます。

```
cfdurl <cfdurl>
```

◇ cfdurl セクション

- ✓ cfdurl セクションの複数定義はできません。

◇ <cfdurl> (ダウンロード URL)

- ✓ 存在する URL である必要があります。
- ✓ 定義できるダウンロード URL は最大1個です。
- ✓ ダウンロード URL に日本語文字は使用できません。
- ✓ ダウンロード URL を””で囲みます。

例) ダウンロード URL として「http://www.sample.co.jp/3Dsample/sample1.cfd」を使用する場合。

```
cfdurl “http://www.sample.co.jp/3Dsample/sample1.cfd “
```

### S.3.3.13. cfdtime セクション

cfdtime セクションでは、マチキャラメタデータファイルに記載するマチキャラデータファイルの更新日時情報を指定します。

cfdtime セクションは以下のように定義されます。

```
cfdtime <flag> [ <time> ]
```

- ◇ cfdtime セクション
  - ✓ cfdtime セクションの複数定義はできません。
- ◇ <flag> (設定種別)
  - ✓ 0(手動設定)または1(自動設定)のいずれかを指定します。
- ◇ <time> (日時情報)
  - ✓ マチキャラデータの更新日時を”YYYY/MM/DD/hh/mm/ss”の形式で指定します。
  - ✓ <flag>が自動設定の場合、省略します。

例) 設定種別として手動設定を、日時情報として2010年10月1日12時を指定する場合。

```
cfdtime 0 2010/10/01/12/00/00
```

### S.3.3.14. filename セクション

filename セクションでは、作成するマチキャラデータファイルの名称を設定します。

filename セクションは以下のように定義されます。

```
filename <cfd-file>
```

◇ filename セクション

- ✓ filename セクションの複数定義はできません。
- ✓ filename セクションを省略することはできません。

◇ < cfd-file > (マチキャラデータファイル)

マチキャラデータファイル名を指定します。

- ✓ 定義できるマチキャラデータファイル数は1個です。
- ✓ マチキャラデータファイル名を""で囲みます。
- ✓ ファイル名に漢字は使用できますが、[特殊文字](#)は使用できません。

例) マチキャラデータファイル名として「sample.cfd」を使用する場合。

```
filename "sample.cfd"
```

### S.3.3.15. マチキャラメイクファイル設定例

マチキャラメイクファイルの設定例を以下に示します。

```
! Sample mascot make file
sample {
    version 5.00;
    distr 0;
    id "SAMPLE";
    title "3D-Sample";
    url "http://www.aaaa.bbbb.ccccc/co.jp";
    popup sample.pop
    contentid "3DSampleContentCID30000100000001";
    fileid "3DSampleContentFID30000100000001";
    mmdurladdress "http://www.sample.co.jp/contents";
    mmdurlquery "content?type=1&id=1";
    mascot {
        behavior mascot1.bhv;
        useful 0 0 0 0 0;
        preview preview1.jpg;
        subtitle "初期"
        cg 0;
        mask_col 0xff0000;
        pers      70 20 2000;
        view      1024 4096 0 0 100 1024 0 0 -4096;
        light 1 100 -100 100 3072 1024;
    }
    mascot {
        behavior mascot2.bhv;
        useful 1 0 6000 3600 10 10;
        preview preview2.bmp;
        subtitle "差替え1"
        cg 0;
        mask_col 0xff0000;
        pers      70 20 2000;
        view      1024 4096 0 0 100 1024 0 4096 4096;
        light 0;
    }
    mascot {
        behavior mascot3.bhv;
        useful 1 0 12000 7200 25 25;
        preview preview3.gif;
        subtitle "差替え2"
        cg 0;
        mask_col 0xff0000;
        pers      70 20 2000;
        view      1024 4096 0 0 -100 1024 0 -4096 -4096;
        light 1 100 100 100 3072 1024;
    }
}
```



```

cfdurl "http://www.sample.co.jp/3Dsample/sample1.cfd";      ...⑬
cfdtime 1;                                                    ...⑭
filename "sample_001.cfd";                                    ...⑮
}

```

- ① 再配布識別子を「再配布不可」に設定します。
- ② マスコットIDに「SAMPLE」を設定します。
- ③ マスコットタイトルに「3D-Sample」を設定します。
- ④ 更新先URLに「http://www.aaaa.bbbb.ccccc/co.jp」を使用します。
- ⑤ ポップアップデータファイルとして「sample.pop」を使用します。
- ⑥ コンテンツIDに「3DSampleContentCID30000100000001」を設定します。
- ⑦ ファイルIDに「3DSampleContentFID30000100000001」を設定します。
- ⑧ MMDファイル取得先URLアドレス部に「http://www.sample.co.jp/contents」を使用します。
- ⑨ MMDファイル取得先URLクエリ部に「content?type=1&id=1」を使用します。
- ⑩ 初期マスコットとして以下のとおり設定します。
  1. ビヘイビアファイル名として「mascot1.bhv」を使用します。
  2. 使用期間を「指定なし」に設定します。
  3. プレビューイメージファイル名として「preview1.jpg」を使用します。
  4. マスコットサブタイトルとして「初期」を使用します。
  5. 2D／3Dフラグを「3D」に設定します。
  6. 透過色として「赤」を使用します。
  7. 透視投影情報を以下のように設定します。
 

・<angle>	70
・<near-clip>	20
・<far-clip>	2000
  8. 視野情報を以下のように設定します。
 

・<up-x>	1024
・<up-y>	4096
・<up-z>	0
・<pos-x>	0
・<pos-y>	100
・<pos-z>	1024
・<look-x>	0
・<look-y>	0
・<look-z>	-4096
  9. 平行光源を以下のように設定します。
 

・<light-id>	1
・<x><y><z>	100 -100 100
・<light>	3072
・<ambient>	1024
- ⑪ 1番目の差し替えマスコットとして以下のとおり設定します。
  1. ビヘイビアファイル名として「mascot2.bhv」を使用します。
  2. 使用期間を「累積経過時間が6000秒、積算通話時間が3600秒、メール受信数が10件、メール送信数が10件のAND条件」に設定します。
  3. プレビューイメージファイル名として「preview2.bmp」を使用します。
  4. マスコットサブタイトルとして「差し替え1」を使用します。
  5. 2D／3Dフラグを「3D」に設定します。
  6. 透過色として「赤」を使用します。

## 7. 透視投影情報を以下のように設定します。

・<angle>	70
・<near-clip>	20
・<far-clip>	2000

## 8. 視野情報を以下のように設定します。

・<up-x>	1024
・<up-y>	4096
・<up-z>	0
・<pos-x>	0
・<pos-y>	100
・<pos-z>	1024
・<look-x>	0
・<look-y>	4096
・<look-z>	4096

## 9. 平行光源を以下のように設定します。

・<light-id>	0
・<x><y><z>	省略
・<light>	省略
・<ambient>	省略

## ⑫ 2番目の差し替えマスコットとして以下のとおり設定します。

1. ビヘイビアファイル名として「mascot3.bhv」を使用します。
2. 使用期間を「累積経過時間が12000秒、積算通話時間が7200秒、メール受信数が25件、メール送信数が25件のAND条件」に設定します。
3. プレビューイメージファイル名として「preview3.gif」を使用します。
4. マスコットサブタイトルとして「差し替え2」を使用します。
5. 2D／3D フラグを「3D」に設定します。
6. 透過色として「赤」を使用します。
7. 透視投影情報を以下のように設定します。

・<angle>	70
・<near-clip>	20
・<far-clip>	2000

## 8. 視野情報を以下のように設定します。

・<up-x>	1024
・<up-y>	4096
・<up-z>	0
・<pos-x>	0
・<pos-y>	-100
・<pos-z>	1024
・<look-x>	0
・<look-y>	-4096
・<look-z>	-4096

## 9. 平行光源を以下のように設定します。

・<light-id>	1
・<x><y><z>	100 100 100
・<light>	3072
・<ambient>	1024

## ⑬ ダウンロード URL に「http://www.sample.co.jp/3Dsample/sample1.cfd」を使用します。

## ⑭ 更新日時情報を自動設定に設定します。

## ⑮ 作成するマチキャラデータファイル名として「sample\_001.cfd」を使用します。

### S.3.4. バルーンデータファイル

バルーンデータファイルは以下に説明するイメージの定義によって、マチキャラに表示するバルーン情報の定義を行います。テキストファイルとして作成します(拡張子は.bln)。バルーンデータファイル名に[特殊文字](#)を含めることはできません。

バルーンデータファイルの漢字コードはシフトJISで作成します。

ここではバルーンデータファイル内でのセクション定義について説明します。  
バルーンを定義するための構文は以下のように定義されます。

```
pallet {  
    <color> [ <color> ... ]  
}  
image {  
    <image-name> : <pos> <time> <image-file> <message-name>  
                  [ [ <pos> <time> <image-file> <message-name> ] ... ]  
}  
message {  
    <message-name> : <fontsize> <default-fontcolor> "<message>" [ '<fontcolor>...' ]  
}
```

- ◇ 重複定義  
各セクションは重複して定義できません。

### S.3.4.1. pallet セクション

pallet セクションでは、マチキャラで使用するバルーンテキストで使用する文字の色を指定します。

pallet セクションは以下のように定義されます。

```
pallet { <color> [ <color> ... ] }
```

◇ pallet セクション

- ✓ pallet セクションの複数定義はできません。
- ✓ message セクションを定義する場合は pallet セクションを省略することはできません。

◇ < color > (文字色)

バルーンに表示するテキストで使用する色を定義します。

- ✓ 文字色はRGB888形式で定義します。
- ✓ 文字色は赤、緑、青をそれぞれ0～255の間で指定する必要があります。
- ✓ 文字色は省略することができません。
- ✓ 文字色は1マスコットに最大256個定義できます。

### S.3.4.2. image セクション

image セクションでは、マチキャラで使用するバルーンを指定します。

image セクションは以下のように定義されます。

```
image {
  <image-name> : <pos> <time> <image-file> [ <message-name> ]
                [ [ <pos> <time> <image-file> [ <message-name> ] ] ... ]
}
```

◇ image セクション

- ✓ image セクションの複数定義はできません。
- ✓ image セクションは最低1個設定が必要です。

◇ < image-name > (バルーン名)

バルーン名を定義する文字列です。

- ✓ [特殊文字](#)、[トークン](#)は使用できません。
- ✓ 16バイト以内で指定します。
- ✓ 数字のみで構成されるバルーン名は指定できません。

◇ < pos > (表示位置)

マスコットに対するバルーン画像の表示位置を指定します。

- ✓ 表示位置は1～4の間で指定する必要があります。

<pos>設定値

表示位置	左上	右上	右下	左下
設定値	1	2	3	4

- ✓ <image-file>を指定する場合は省略できません。

◇ < time > (表示時間)

バルーン画像の表示時間を指定します。

- ✓ 表示時間はミリ秒単位で指定します。
- ✓ 1つのバルーンと同じ表示位置に複数のバルーン画像を設定する場合は、1以上の値を設定する必要があります。
- ✓ 表示時間を指定しない場合は0ミリ秒とします。
- ✓ <image-file>を指定する場合は省略できません。

◇ < image-file > (バルーン画像ファイル)

バルーン画像ファイルを指定します。

- ✓ バルーン画像ファイルは存在するファイルである必要があります。
- ✓ ファイル名に漢字は使用できますが、[特殊文字](#)は使用できません。
- ✓ バルーン画像ファイル名に[特殊文字](#)を除く[トークン](#)を使用する場合は二重引用符で囲む必要があります。
- ✓ 1つのバルーンに定義できるバルーン画像数は最大32です。
- ✓ バルーンの全ての表示位置に画像が設定されている必要があります。
- ✓ <pos>または<time>を指定する場合は省略できません。

◇ < message-name > (テキスト名)

バルーンに表示するテキスト名を指定します。

- ✓ テキスト名は存在するテキスト名である必要があります。
- ✓ [特殊文字](#)、[トークン](#)は使用できません。
- ✓ 16バイト以内で指定します。
- ✓ 1つのバルーン画像ファイルに最大1個のテキスト名を定義することができます。
- ✓ バルーンにテキストを表示しない場合は省略します。

### S.3.4.3. message セクション

message セクションでは、バルーンに表示するメッセージを指定します。

message セクションは以下のように定義されます。

```
message {
  <message-name> : <fontsize> <default-fontcolor> "<message>" [ '<fontcolor>...' ]
}
```

◇ message セクション

- ✓ message セクションの複数定義はできません。

◇ < message-name > (テキスト名)

テキスト名を定義する文字列です。

- ✓ 特殊文字、トークンは使用できません。
- ✓ 16バイト以内で指定します。

◇ < fontsize > (文字サイズ)

メッセージ名に続き、:(コロン)の次にバルーン表示するテキストの文字サイズを指定します。

- ✓ 本バージョンでは12、18、24、36、48および72のいずれかを指定します。単位は dot です。
- ✓ 文字サイズは1メッセージに1つ定義することができます。

◇ < default-fontcolor > (通常文字色)

バルーンに表示するテキストの基本の色のインデックスを指定します。

- ✓ 通常文字色を省略することはできません。
- ✓ 通常文字色は1メッセージに1つ定義することができます。
- ✓ pallet で設定された順番-1の値を指定します。

◇ < message > (メッセージ)

バルーンに表示するテキストを指定します。

- ✓ メッセージを省略することはできません。
- ✓ メッセージを"'"で囲みます。
- ✓ 改行コードを含むメッセージを設定する場合は、改行コードを「\n」と記述します。
- ✓ 改行コードは1文字として扱います。
- ✓ メッセージの文字数が表示可能文字数を超える場合は、超過分は表示しません。

表示可能文字数は以下のとおりです。

ターゲット VRAM エリア	QCIF サイズ	QVGA サイズ	VGA サイズ	XGA サイズ
文	12dot	3.5 文字×1 行	7 文字×2 行	14 文字×4 行
字	18dot	非対応	非対応	非対応
サ	24dot	非対応	3.5 文字×1 行	7 文字×2 行
イ	36dot	非対応	非対応	非対応
ズ	48dot	非対応	非対応	3.5 文字×1 行
	72dot	非対応	非対応	3.5 文字×1 行

※表示文字数は全角文字を1文字、半角文字を0.5文字とします。

- ✓ 改行コードも他の文字と同じ方法で文字色を設定します。  
 <fontcolor>で文字色が指定されている場合は、<fontcolor>で指定されている色で表示します。  
 <fontcolor>で文字色が指定されていない場合は、通常文字色で指定された色で表示します。  
 改行コードは文字として表示されないため、移動機端末上での確認はできません。
- ✓ 文字列置換機能を利用した文字列を設定する場合は文字列置換機能非対応機種用文字列を記述したあと、制御コードを記述し、文字列置換機能対応機種用文字列を記述します。
- ✓ 制御コードには以下の種類があります。

制御	制御コード	制御	制御コード
repace 制御	0xDBFF 0xDFF1	End 制御	0xDBFF 0xDFF3
break 制御	0xDBFF 0xDFF2	noparagraph 制御	0xDBFF 0xDFF5

- ✓ 制御コードは2文字として扱います。
- ✓ 制御コードも他の文字と同じ方法で文字色を設定します。  
 <fontcolor>で文字色が指定されている場合は<fontcolor>で指定されている色で表示します。  
 <fontcolor>で文字色が指定されていない場合は通常文字色で指定された色で表示します。  
 制御コードは文字として表示されないため、移動機端末上での確認はできません。
- ✓ 文字列置換機能対応機種用文字列に設定する置換文字列は表示位置に置換コードを記述します。  
 置換コードは NTT ドコモおよび各移動機メーカーが取り決めます。
- ✓ 置換コードは2文字として扱います。
- ✓ 置換コードも他の文字と同じ方法で文字色を設定します。  
 <fontcolor>で文字色が指定されている場合は<fontcolor>で指定されている色で表示します。  
 <fontcolor>で文字色が指定されていない場合は通常文字色で指定された色で表示します。  
 ただし、実際に移動機に表示する際は、置換された文字列全てを置換コードの1文字目に設定されている文字色で表示します。
- ✓ メッセージに絵文字を使用する場合はNTTドコモ提供の絵文字入力ソフト「i絵文字」を使用する必要があります。
- ✓ メッセージは1テキストに1つ定義することができます。

◇ <fontcolor> (文字色)

バルーンに表示するテキストで、通常文字色と異なる色を使用する文字および文字色のインデックスを指定します。

- ✓ 書式は“font-no:font-color”です。
- ✓ font-no はテキスト中の何番目の文字かを指定します。
- ✓ 連続する複数の文字を同色で表示する場合は font-no を-(ハイフン)で繋いで指定することができます。
- ✓ font-color は pallet で設定された順番-1の値を指定します。
- ✓ 文字色を' 'で囲みます。
- ✓ テキストを1色で表示する場合は省略することができます。



### S.3.4.4. バルーンデータファイル設定例

バルーンデータファイルの設定例を以下に示します。

```
pallet {                                     ... ①
    0x000000;
    0xff0000;
    0x00ff00;
}
image {
    bal1 : 1 0 image1.gif mess1;           ... ②
           2 0 image2.gif mess1;
           3 0 image3.gif mess1;
           4 0 image4.gif mess1;
    bal2 : 1 300 image5.gif mess2;         ... ③
           1 400 image6.gif mess3;
           1 500 image7.gif mess4;
           2 0 image8.gif mess5;
           3 0 image9.gif mess5;
           4 0 image10.gif mess5;
}
message {
    mess1 : 24 0 "message1" '1-2:1 8:2'   ... ④
    mess2 : 24 1 "me"                     ... ⑤
    mess3 : 24 0 "ssage"                   ... ⑥
    mess4 : 24 2 "2"                       ... ⑦
    mess5 : 12 0 "message3"                ... ⑧
    mess6 : 12 0 "message3DBFFDFF1 今日の天気は DBFFDFFD3 です。
DBFFDFF2DBFFDFF3"                        ... ⑨
    mess7 : 12 0 "message3DBFFDFF1 今日の天気は DBFFDFF5DBFFDFFD3 です。
DBFFDFF2DBFFDFF3"                        ... ⑩
}
```

#### ① 文字色

バルーンに表示する文字には以下の色を使用します。

- ・0x000000
- ・0xff0000
- ・0x00ff00

#### ② bal1 イメージが指定された場合のバルーン表示は以下のようになります。

1. バルーンがマスコットの左上に表示される場合は「image1.gif」を表示する。
2. バルーンがマスコットの右上に表示される場合は「image2.gif」を表示する。
3. バルーンがマスコットの右下に表示される場合は「image3.gif」を表示する。
4. バルーンがマスコットの左下に表示される場合は「image4.gif」を表示する。
5. いずれの位置に表示される場合も、テキストは mess1 メッセージが表示される。
6. 他のイメージが指定されるかバルーン無効のイベントが発生するまで、同じ画像ファイルを使用する。

#### ③ bal2 イメージが指定された場合のバルーン表示は以下のようになります。

1. バルーンがマスコットの左上に表示される場合は最初の300ミリ秒間は「image5.gif」を表示する。その後の400ミリ秒間は「image6.gif」、その後の500ミリ秒間は「image7.gif」を表示する。「image7.gif」を500ミリ秒間表示した後は

- 「image5.gif」に戻り、同様の表示を繰り返す。
2. 「image5.gif」が表示されている間は mess2 メッセージを表示し、「image6.gif」が表示されている間は mess3 メッセージ、「image7.gif」が表示されている間は mess4 メッセージを表示する。
  3. バルーンがマスコットの右上に表示される場合は「image8.gif」を表示する。
  4. バルーンがマスコットの右下に表示される場合は「image9.gif」を表示する。
  5. バルーンがマスコットの左下に表示される場合は「image10.gif」を表示する。
  6. 表示位置が右上、右下および左下の場合は、テキストは mess5 メッセージが表示される。
- ④ mess1 メッセージが指定された場合のテキスト表示は以下のようになります。
1. バルーンに24dot の文字サイズで「message1」と表示する。
  2. テキストは1文字目と2文字目(「me」)は赤色、8文字目(「1」)は緑色、その他の文字は黒色で表示する。
- ⑤ mess2 メッセージが指定された場合のテキスト表示は以下のようになります。
1. バルーンに24dot の文字サイズで「me」と表示する。
  2. テキストは赤色で表示する。
- ⑥ mess3 メッセージが指定された場合のテキスト表示は以下のようになります。
1. バルーンに24dot の文字サイズで黒色で「ssage」と表示する。
  2. テキストは黒色で表示する。
- ⑦ mess4 メッセージが指定された場合のテキスト表示は以下のようになります。
1. バルーンに24dot の文字サイズで黒色で「2」と表示する。
  2. テキストは緑色で表示する。
- ⑧ mess5 メッセージが指定された場合のテキスト表示は以下のようになります。
1. バルーンに12Dot の文字サイズで黒色で「message3」と表示する。
  2. テキストは黒色で表示する。
- ⑨ mess6 メッセージが指定された場合のテキスト表示は以下のようになります。
1. 文字列置換非対応機種ではバルーンに12dot の文字サイズで「message3」と表示する。
  2. 文字列置換対応機種ではバルーンに12dot の文字サイズで「今日の天気はxxです。」と表示する。  
(xxは携帯端末内で編集)
  3. 「xx」はオート改行ありで表示する。
  4. テキストは黒色で表示する。
- ⑩ mess7 メッセージが指定された場合のテキスト表示は以下のようになります。
1. バルーンに12dot の文字サイズで「message3」と表示する。
  2. 文字列置換対応機種ではバルーンに12dot の文字サイズで「今日の天気はxxです。」と表示する。  
(xxは携帯端末内で編集)
  3. 「xx」はオート改行なしで表示する。
  4. テキストは黒色で表示する。

### S.3.5. 定型移動データファイル

定型移動データファイルは以下に説明する定型移動の定義によって、マチキャラの移動ルート  
の定義を行います。テキストファイルとして作成します（拡張子は.fmf）。1つの定型移動データフ  
ァイルに複数の定型移動を定義することができます。

定型移動データファイル名に**特殊文字**を含めることはできません。

定型移動データファイルの漢字コードはシフトJISで作成します。

ここでは定型移動データファイル内での定義について説明します。

定型移動を定義するための構文は以下のように定義されます。

```
<route-name> {
    <point-count>
    [[<x> <y>] ... ]
}
```

- ◇ <route-name> (定型移動データ名)  
定型移動データ名を定義する文字列です。
  - ✓ **特殊文字**、**トークン**は使用できません。
  - ✓ 1つの定型移動ルートに1つの定型移動データ名を定義することができます。
  - ✓ 定型移動データ名は重複して定義することはできません。
  - ✓ 定型移動データ名は省略することはできません。
- ◇ <point-count> (ポイント数)  
移動ルートポイント数を設定します。
  - ✓ ポイント数は1以上を設定する必要があります。
  - ✓ ポイント数は省略することはできません。
- ◇ <x><y> (X座標、Y座標)  
定型移動ルートポイント位置をX座標とY座標で指定します。
  - ✓ X座標とY座標は空白で区切ります。
  - ✓ <point-count>分の<x><y>を指定する必要があります。
  - ✓ <x><y>は0～639の間で指定する必要があります。
  - ✓ X座標およびY座標は省略することはできません。

#### S.3.5.1. 定型移動データファイル設定例

定型移動データファイルの設定例を以下に示します。

```
fixedmove1{      ←定型移動データ名
    3             ←ポイント数
    10 10        ←ポイント1の座標
    40 90        ←ポイント2の座標
    100 15       ←ポイント3の座標
}
```

### S.3.6. ポップアップデータファイル

ポップアップデータファイルは以下に説明するイメージの定義によって、マチキャラに表示するポップアップ情報の定義を行います。テキストファイルとして作成します（拡張子は.pop）。

ポップアップデータファイル名に**特殊文字**を含めることはできません。

バルーンデータファイルの漢字コードはシフトJISで作成します。

ここではポップアップデータファイル内での定義について説明します。

ポップアップを定義するための構文は以下のように定義されます。

```
[ <event-id> [
  <action-type> : <popup-name>;
  [ [<action-type> : <popup-name>] ... ]
]]
pop{
  <popup-name> : <size> <popup-file> <font-color>;
  [ [ <size> <popup-file> <font-color>; ]....]
  [ [ <popup-name> : <size> <popup-file> <font-color>;
    [ [<size> <popup-file> <font-color>; ]....] ]... ]
}
```

◇ <event-id> (イベントコード)

ポップアップを設定するイベントコードを指定します。

- ✓ イベントコードは省略することはできません。
- ✓ イベントコードは重複して定義することはできません。
- ✓ イベントコードに設定できる文字はアルファベット1文字です。

◇ <action-type> (アクション種別)

アクション種別を指定します。

- ✓ アクション種別は省略することはできません。
- ✓ アクション種別は重複して定義することはできません。
- ✓ アクション種別の設定値は NTT ドコモサービス仕様で規定された内容に準じます。

◇ <popup-name> (ポップアップ名)

ポップアップ名を定義する文字列です。

- ✓ **特殊文字**、**トークン**は使用できません。
- ✓ 16バイト以内で指定します。
- ✓ ポップアップ名は重複して定義することはできません。
- ✓ ポップアップ名は省略することはできません。
- ✓ 1つのポップアップに最大3つのポップアップ画像を設定することができます。
- ✓ 数字のみで構成されるポップアップ名は指定できません。

◇ <size> (サイズ種別)

ポップアップ画像のサイズ種別を指定します

- ✓ サイズ種別は省略することはできません。
- ✓ 1つのアクション種別に同じサイズ種別を指定することはできません。
- ✓ サイズ種別の設定値は以下のとおりです。

サイズ種別の設定値

ターゲット VRAM エリア	QVGA サイズ	VGA サイズ	XGA サイズ
ポップアップ画像サイズ	288 × 186pixel	456 × 372pixel	684 × 558pixel
設定値	1	2	3

- ☆ <popup-file> (ポップアップ画像ファイル)  
ポップアップ画像ファイルを指定します
  - ✓ ポップアップ画像ファイルは存在するファイルである必要があります。
  - ✓ ファイル名に漢字は使用できませんが、[特殊文字](#)は使用できません。
  - ✓ ポップアップ画像ファイル名に[特殊文字](#)を除く[トークン](#)を使用する場合は二重引用符で囲む必要があります。
  - ✓ ポップアップ画像ファイルは省略することはできません。
- ☆ <font-color> (表示文字色)  
ポップアップに表示する文字の色を指定します。
  - ✓ 表示文字色はRGB888形式で定義します。
  - ✓ 表示文字色は赤、緑、青をそれぞれ0～255の間で指定する必要があります。
  - ✓ 表示文字色は省略することはできません。

### S.3.6.1. ポップアップデータファイル設定例

ポップアップデータファイルの設定例を以下に示します。

```

W {                                     ... ①
    05 : basic;
}
X {                                     ... ②
    07 : basic;
}
Y {                                     ... ③
    12 : emergency;
}
A {                                     ... ④
    00 : basic;
    01 : basic;
    10 : emergency;
    15 : schedule;
}
Z {                                     ... ⑤
    15 : schedule;
}
pop {
    basic : 1 pop0_0.gif 0x000000;      ... ⑥
           2 pop0_1.gif 0x000000;
           3 pop0_2.gif 0x000000;
    emergency : 1 pop1_0.gif 0xff0000;   ... ⑦
                2 pop1_1.gif 0xff0000;
                3 pop1_2.gif 0xff0000;
    schedule : 1 pop2_0.gif 0x0000ff;    ... ⑧
                2 pop2_1.gif 0x00ff00;
                3 pop2_2.gif 0x0000ff;
}

```

- ① イベントコード W(アラームイベント)でアクション種別毎に使用するポップアップは以下のようになります。
  1. アクション種別05の場合は「basic」を使用する。
- ② イベントコード X(ダイアログイベント)でアクション種別毎に使用するポップアップは以下のようになります。
  1. アクション種別07の場合は「basic」を使用する。
- ③ イベントコード Y(i コンシェル TOP ショートカットフォーカスイベント)でアクション種別毎に使用するポップアップは以下のようになります。
  1. アクション種別12の場合は「emergency」を使用する。
- ④ イベントコード A(i コンシェルイベント)でアクション種別毎に使用するポップアップは以下のようになります。
  1. アクション種別00の場合は「basic」を使用する。
  2. アクション種別01の場合は「basic」を使用する。
  3. アクション種別10の場合は「emergency」を使用する。

4. アクション種別15の場合は「schedule」を使用する。
- ⑤ イベントコード Z(i コンシェル TOP イベント)でアクション種別毎に使用するポップアップは以下のようになります。
1. アクション種別15の場合は「schedule」を使用する。
- ⑥ basic ポップアップが指定された場合のポップアップ表示は以下のようになります。
1. ターゲット VRAM エリアが QVGA サイズの場合は「pop0\_0.gif」を表示し、文字列は黒色で表示する。
2. ターゲット VRAM エリアが VGA サイズの場合は「pop0\_1.gif」を表示し、文字列は黒色で表示する。
3. ターゲット VRAM エリアが XGA サイズの場合に「pop0\_2.gif」を表示し、文字列は黒色で表示する。
- ⑦ emergency ポップアップが指定された場合のポップアップ表示は以下のようになります。
1. ターゲット VRAM エリアが QVGA サイズの場合は「pop1\_0.gif」を表示し、文字列は赤色で表示する。
2. ターゲット VRAM エリアが VGA サイズの場合は「pop1\_1.gif」を表示し、文字列は赤色で表示する。
3. ターゲット VRAM エリアが XGA サイズの場合に「pop1\_2.gif」を表示し、文字列は赤色で表示する。
- ⑧ schedule ポップアップが指定された場合のポップアップ表示は以下のようになります。
1. ターゲット VRAM エリアが QVGA サイズの場合は「pop2\_0.gif」を表示し、文字列は青色で表示する。
2. ターゲット VRAM エリアが VGA サイズの場合は「pop2\_1.gif」を表示し、文字列は緑色で表示する。
3. ターゲット VRAM エリアが XGA サイズの場合に「pop2\_2.gif」を表示し、文字列は青色で表示する。