

# テキストベース人間関係シミュレーション 統合仕様書

---

## 1.はじめに

---

### 1.0 本仕様書の目的

本仕様書は、本シミュレーションアプリケーションの機能、構造、および内部処理の設計内容を明示することを目的とする。本アプリは、複数のキャラクター同士の関係性や会話の推移を、リアルタイムで観察・管理する形式のテキストベースのシミュレーションである。

プレイヤーは「管理人」として、媒体越しにキャラクターたちのやり取りを見守りながら、必要に応じて特定のイベントに介入する役割を担う。

---

### 1.1 システム全体概要

本アプリは主に以下の要素から構成される：

- UI画面**：キャラクターの様子、ログ、イベントなどを確認・管理するための画面群
- キャラクター管理**：性格・関係性・状態の登録と編集
- イベント生成・表示**：時間帯や性格等に応じた会話イベントの自動発生
- 信頼度・好感度・ラベル変化処理**：関係や感情の変化を制御する機構
- ログと日報**：会話やラベル変化を記録し、振り返りを可能とする
- セーブ・ロード機能**：プレイ内容の保存・復元を行う

本アプリはWebベースのローカルアプリケーションとして設計されており、インターネット接続を必要とせず、単体で動作する。

---

### 1.2 用語の定義

- **キャラクター**：プレイヤーが管理・観察対象とする仮想存在。複数存在し、互いに関係を持つ。
  - **プレイヤー**：キャラクターたちに干渉可能な上位存在。主に困り事相談への対応などで介入する。
  - **関係ラベル**：キャラクター同士の関係性（例：なし／認知／友達／親友／恋人／家族）を示す。
  - **感情ラベル**：キャラクターA→Bにおける一時的な印象や感情（例：気になる／嫌いかも／気まずいなど）を示す。
  - **信頼度**：キャラクター→プレイヤー間に存在する単一方向の数値指標。
  - **好感度**：キャラクターA→B間に存在する単一方向の数値指標。
- 

### 1.3 本仕様書の構成

本仕様書は以下の章構成で構成される：

- **第1章：はじめに ...** システム全体の概要と用語の整理
  - **第2章：UI仕様 ...** 各画面の構成とUI要素の設計
  - **第3章：キャラクター仕様 ...** キャラクター同士の関係・感情の構造と変化
  - **第4章：イベント仕様 ...** 会話イベントの分類と仕様
  - **第5章：内部処理・抽選仕様 ...** ラベル変化・好感度変化・抽選処理の詳細
  - **第6章：データ構造と保存仕様 ...** 内部データの構成と保存・読み込み方式
  - **第7章：MBTI診断仕様と性格タイプ決定**
-

## 2.UI仕様

### 2.1 メイン画面 UI設計仕様書（再整理版）

#### 2.1.1 画面の目的

- キャラクターの状態観察とログ閲覧を主目的とした、アプリの中心ビュー。
- 管理室や日報など、他機能への導線を兼ねる。
- SPA的構造でありながら、UI上はページ遷移風の構成。

#### 2.1.2 画面構成概要（全体）

[管理室] [日報] [時刻表示] [日付表示]

▼ みんなの様子（キャラ一覧・横スクロール）

キャラA	キャラB	キャラC	キャラD
------	------	------	------

 → スクロール可

※クリックで詳細ステータスへ

▼ 困りごと相談エリア

・彩花から相談があります ▶ [対応する]

・志音が困っているようです ▶ [対応する]

[+ 相談をさらに受ける]

▼ ログ表示エリア（CLI風）

[08:42] EVENT: 莉音と咲が会話中...

莉音「ねえ、あの本、読んだ？」

咲「えっ、まだだけど、面白かった？」

[08:45] SYSTEM: 信頼度が上昇しました。

.....

#### 2.1.3 各構成要素の仕様

##### ◆ 上部メニュー

項目	説明
管理室	キャラの追加・編集・削除用ビューへの導線
日報	日別・キャラ別イベントを確認できる画面
時刻・日付	右上にリアルタイムで表示（現実時間と同期）

◆ みんなの様子（キャラ一覧）

項目	説明
表示形式	横スクロール可能なカード型表示
表示内容	キャラ名（必須）＋状態（例：「風邪」「就寝中」などテキスト）
遷移機能	各カードクリックでキャラステータス画面に遷移

◆ 困りごと相談エリア（プレイヤー介入イベント）

項目	内容
表示位置	ログ表示エリアの上、キャラ一覧の下に配置
表示内容	キャラ名と簡易相談文、「対応する」ボタン（1イベント1行）＋追加ボタン
初期表示件数	最大2件、明示操作により最大3件まで
消滅処理	約1時間で未対応の相談は自動的に非表示
遷移後動作	回答ポップアップ（別仕様：2.4節）を開き、回答完了後にキャラの返答を表示
ログ出力	回答完了時にログへ反映（セリフは非表示）

◆ ログ表示エリア（CLI風）

項目	内容
表示スタイル	CLI風（タイピング演出＋色分け）
ログ形式	EVENT: 会話・行動など／SYSTEM: 信頼度や感情変化
既読処理	未読があるとき「未読あり」表示、スクロールで既読化

## 2.2 キャラステータス画面

### 2.2.1 概要

- キャラクターの現在状態・関係性・性格パラメータを閲覧可能な専用ビュー。
- 編集機能は持たず、閲覧専用である。
- キャラ一覧（みんなの様子）から遷移する形で表示。

### 2.2.2 画面構成（例）

————— ステータス画面（キャラA） —————

名前：碧  
性格タイプ：INFP  
話し方：くだけた（一人称：俺、語尾：～じゃん）  
現在状態：風邪

▼ 性格パラメータ

社交性：■■■■□      気配り：■■■■□□  
頑固さ：■■□□□      行動力：■■■■■  
表現力：■■■■□

▼ 関係一覧（横スクロール・簡易表示）

彩花：関係ラベル「友達」 | 好感度 [■■■■□] | 気になる  
志音：関係ラベル「認知」 | 好感度 [■■□□□] | なし  
（クリックで個別詳細ページへ遷移）

▼ イベント履歴（最近5件）

- 6/29 彩花と会話した
- 6/28 志音とすれ違った
- 6/28 志音と会話した
- 6/27 彩花と会話した
- 6/27 志音と会話した

2.2.3 詳細仕様

◆ 基本情報

項目	表示形式
名前	太字表示
MBTI	タイプ名（例：INFP）
話し方	プリセット名＋カスタム欄（例：「丁寧（私、ですわ）」）
現在状態	テキスト（例：「就寝中」「風邪」「活動中」など）

◆ 性格パラメータ

- 5段階バー表示（視覚的にスライダーまたはブロック表示）
- 対象項目：

- 社交性
- 気配り傾向
- 頑固さ
- 行動力
- 表現力

#### ◆ 関係一覧

項目	内容
表示形式	横スクロール型のキャラカード
表示内容	対象キャラ名、関係ラベル、好感度バー、感情ラベル
遷移機能	各キャラをクリックすると、個別関係詳細画面へ遷移

#### ◆ イベント履歴

項目	内容
表示件数	最大5件まで
表記形式	「日付＋対象キャラ＋概要」例：「6/29 彩花と会話した」
補足	詳細な会話内容等は表示せず、ログ画面と分離する

### 2.2.4 備考

- 本画面は情報表示専用であり、編集操作は不可。
- キャラの編集は「管理室」画面から行う（別章に記述）。
- 関係ラベルや感情ラベルの定義はイベントや内部処理仕様を参照。

## 2.3 キャラ関係詳細画面

### 2.3.1 画面の目的

- キャラクターA⇄Bの相互関係性の詳細表示を行う専用ビュー。

- 「関係ラベル」「好感度（片方向）」「感情ラベル（片方向）」を左右対称で見やすく提示。
- 直近の関係イベント履歴を**自然文形式で簡易的に表示**し、状態変化の参照を可能にする。

### 2.3.2 表示構成（UIイメージ）

————— キャラA⇄キャラB 関係詳細 —————	
キャラA：碧	⇄ キャラB：彩花
関係ラベル：友達	
好感度（碧→彩花）：■■■■□	
感情ラベル（碧→彩花）：気になる	
好感度（彩花→碧）：■■■■■	
感情ラベル（彩花→碧）：好きかも	
▼ 最近のやり取り履歴（簡易表示）	
・ 6/29 彩花と会話	
・ 6/27 碧と過ごした	
・ 6/24 初対面	

### 2.3.3 項目仕様

#### ◆ キャラクター基本情報

表示項目	内容
キャラ名（A/B）	画面の左右に表示。ステータス画面からの遷移時に確定済み。
関係ラベル	「なし」「認知」「友達」など、 <b>双方向同値ラベル</b> 。
好感度（A→B）	数値（-100～+100）＋バー表示。
感情ラベル（A→B）	「気になる」「好きかも」「気まずい」など。片方向の感情状態を表示。

※左右対称に、A→B、B→A 両方向が明示される。

#### ◆ 関係履歴表示



項目	内容
内容	直近の関係イベントを自然文で3～5件表示
表記例	「6/29 彩花と会話した」「6/27 碧と過ごした」
非表示項目	発端キャラ・イベント種別などの内部分類情報は非表示
編集不可	閲覧専用。ログ編集や詳細展開はこの画面には実装しない

#### 2.3.4 備考・遷移元など

- この画面はキャラステータス画面内の「関係一覧」からアクセス。
- 編集機能は持たず、完全に表示専用ビューとする。

## 2.4 プレイヤー介入イベント UI設計仕様書（再整理版）

### 2.4.1 画面の目的

- キャラクターからプレイヤーへの「困りごと」相談を提示し、1回限りの介入を可能にするUIである。
- 回答内容によって信頼度が変動し、結果はログに反映される。
- UI構成上はメイン画面内の一部コンポーネントとして実装される。

### 2.4.2 メイン画面内での配置と表示条件

項目	内容
表示位置	メイン画面中央部、「みんなの様子」下、「ログ表示エリア」上
表示条件	イベントが存在する場合のみ（最大3件まで）
初期表示件数	初回：2件（追加はボタンにて）
自動消滅	未対応相談は約1時間で自動的に非表示となる

### 2.4.3 表示構成（概要エリア）

#### ▼ 困りごと相談エリア

- ・彩花から相談があります ▶ [対応する]
- ・志音が困っているようです ▶ [対応する]
- [+ 相談をさらに受ける]

- 「+相談をさらに受ける」クリックで追加表示（最大3件まで）

### 2.4.4 回答ポップアップ構成

#### 【共通構成】

——— 困りごと ———  
彩花「〇〇を一したくて、どうしたらいいかな？」

#### ▼ 回答形式（選択肢 or 穴埋め）

##### [形式A] 選択肢式

- ・A：とりあえずやってみたら？
- ・B：誰かに相談してみるのはどう？
- ・C：ちょっと休むのもありだと思うよ

##### [形式B] 穴埋め式（1文定型：空欄1つ）

「\_\_\_\_\_してみたら？」

[送信する]（一度のみ・確定後は再回答不可）

- 回答終了後はキャラの自然な返答（「ありがとう！」など）を表示し、ポップアップを閉じる。

### 2.4.5 ログ出力仕様

- 回答結果はログに出力される（会話内容やセリフは含まない）：

[10:21] EVENT：彩花がプレイヤーに相談しています...  
[10:23] SYSTEM：彩花からの信頼度が上昇しました。

- 表示対象：イベント発生・信頼度変動などの「事実」

- 表示対象外：セリフ本文・感情表現・プレイヤー回答内容

#### 2.4.6 その他補足

- 回答形式は相談テンプレートごとに指定されており、選択式または穴埋め式のいずれか。
- 相談イベントは履歴上はログにのみ記録され、日報や画面上での個別記録は行わない。
- 信頼度以外の内部パラメータ（感情ラベル、関係ラベル）には影響しない。

---

## 2.5 日報画面

### 2.5.1 概要

- 日付ごとに、その日に発生したイベントやラベル変化を一覧で確認できる画面。
- 「発生イベント一覧」と「ラベル変化履歴」を分けて表示。
- フィルターにより、日付・キャラクター・変化種別で絞り込み可能。

### 2.5.2 画面構成イメージ

——— 日報（2025/06/29） ———

▼ フィルター

[日付：2025/06/29] [キャラ選択 ▼] [変化タイプ ▼]

▼ 発生イベント一覧（簡易表示）

・ 10:03 彩花と碧が会話した

・ 15:42 志音が響を見かけた

▼ ラベル変化履歴

・ 碧→彩花：感情ラベル「なし」→「気になる」

・ 詠翔→響：関係ラベル「なし」→「友達」

### 2.5.3 機能詳細

- ◆ フィルター

項目	内容
日付	カレンダーUIまたは前日/翌日ボタンで切替
キャラ選択	チェックボックスまたはドロップダウン（複数選択可）
変化タイプ	「関係ラベル」「感情ラベル」「イベント」のいずれかを選択可能

#### ◆ 発生イベント一覧

- 時刻＋キャラクター名を使った簡易記述（最大10件、スクロール可）
- イベントの種類は非表示（詳細ログ画面へ遷移可能）
- 表記例：
  - [10:03] 彩花と碧が会話した
  - [15:42] 志音と響が挨拶した

#### ◆ ラベル変化履歴

- 感情ラベルは一方向で表示（例：「碧→彩花」）
- 関係ラベルは双方向をまとめて表記（例：「詠翔⇄響」）
- 表記例：
  - 碧→彩花：印象「なし」→「気になる」
  - 詠翔⇄響：関係「なし」→「友達」

#### 2.5.4 備考

- 初期表示は「当日（または最新）」の日付で表示
- イベント記録は内部ログと連動
- 各記録をクリックすると、該当ログの詳細ビューに遷移可能（拡張予定）

## 2.6 管理室（キャラクター管理ページ）

2.6.1 概要

- キャラクターの追加・編集・削除、および情報確認を行うためのページ。
- 性格診断や初期関係の設定も含まれ、キャラクター初期化における中核的役割を担う。

2.6.2 画面構成

管理室

[+キャラクター追加]（クリックでフォーム展開）

▼ 既存キャラクター一覧（名前順・操作ボタン付き）

- 葵

[情報]

[編集]

[削除]

- 彩花

[情報]

[編集]

[削除]

2.6.3 キャラクター追加フォーム

項目	入力形式
名前	テキスト入力欄
MBTI	診断スタートボタン（スライダー形式）または選択入力
性格パラメータ	スライダー（1～5）×5項目（社交性／気配り／頑固さ／行動力／表現力）
話し方	ラジオボタン＋自由記述欄（口調・語尾・一人称等）
興味関心ジャンル	テキスト入力（タグ形式も検討）
初期関係設定	他キャラをプルダウン選択＋関係タイプ指定（例：兄弟・親・幼なじみなど）
呼び方設定（任意）	キャラA→B / B→A それぞれにテキスト入力欄（空欄可）
活動傾向	ラジオボタン（朝型／夜型／通常）

- 「追加」ボタンは必須項目がすべて入力された場合に限り有効化される。

#### 2.6.4 各種モード仕様

##### 情報モード

- [情報] ボタンで表示。
- 編集フォームと同じ構成だが全項目が読み取り専用。
- 操作ボタン（保存など）は非表示または非活性化。
- ステータス画面の流用基盤にもなる。

##### 編集モード

- [編集] ボタンで表示。
- すべての項目を編集可能（※「名前」欄は固定または要検討）。
- 既存キャラデータを事前入力済みの状態で表示。

##### 削除モード

- [削除] ボタンで確認ダイアログを表示。
- OK選択でキャラクター削除、キャンセルで操作中止。

#### 2.6.5 備考

- 初期関係設定は、「関係ラベル」「好感度」「感情ラベル」の初期値生成に影響を与える。
- MBTI診断結果をもとに性格パラメータを自動割り当てする機能も検討中。
- 編集時に既に削除されたキャラが関係先に存在する場合、関係の無効化または再設定を促す。

---

## 2.7 ログ画面（CLI風ログ）

### 2.7.1 概要

- 会話・イベントの履歴を「CLI風ログ」として表示する画面。

- EVENT（イベント発生）／SYSTEM（状態変化）を主軸に構成。
- セリフ表示はログ内でインデント・改行あり。
- タイピング風アニメーションを適用。

### 2.7.2 表示内容

- EVENT: イベント（会話など）の開始を示す。
- セリフ: キャラ同士の発言（インデントあり）。
- SYSTEM: 状態変化（感情ラベル・関係ラベル・好感度など）。

```
[08:42] EVENT: 莉音と咲が会話中...
          莉音「ねえ、あの本、読んだ？」
          咲「えっ、まだだけど、面白かった？」
[08:45] SYSTEM: 莉音→咲の好感度が上昇しました。
[08:47] SYSTEM: 莉音→咲の印象が「気になる」に変化しました。
```

### 2.7.3 表示スタイル

- EVENT行／SYSTEM行は色分け・強調表示。
  - SYSTEM行は橙系カラー（例：#FFA500）。
- セリフは改行・字下げあり（キャラ名は省略可）。
- 未読ログがある場合、下部に「未読があります」表示。

### 2.7.4 表示順と並び替え

- ログ表示順は「新しい順」「古い順（時系列順）」を切替可能。
- 初期設定は最新順表示。

### 2.7.5 備考

- ログに表示されるのは概要と結果のみ。
  - セリフありイベントでも詳細プロンプトや選択肢等は記録されない。
- プレイヤーへのフィードバック（信頼度変化など）は SYSTEM 行で簡潔に伝える。

- 数値は表示せず、「～が変化しました」と記載。
-



## 3.キャラクター仕様

---

### 3.1 基本情報と性格プロフィール

キャラクターは、プレイヤーが管理する対象であり、以下のような基本情報と性格パラメータを持つ。

#### 3.1.1 基本情報

- **名前**：キャラクターごとに設定可能な表示名。ゲーム内すべてのUIで使われる。
- **話し方**：口調や文体の設定。プリセット（例：「ていねい」「くれた」）から選択可能で、カスタムも可。会話生成時のトーンに影響。

また、各キャラクターには「活動傾向」として、**朝型**／**夜型**／**通常型**のいずれかを設定する。これはイベント発生時間帯補正に影響し、キャラクターの生活リズムを反映する要素としても扱われる。

#### 3.1.2 MBTIタイプ

- キャラクターの性格傾向を表現するため、16Personalitiesに基づくMBTIタイプ（例：INFP、ESTJなど）を採用。
- MBTIの設定方法は以下の2通りから選択できる：
  - i. **スライダー診断による自動推定** キャラ作成時に提示される5段階スライダー質問（例：「一人の方が落ち着く？」「予定は細かく立てたいタイプ？」など）に回答することで、自動的にMBTIタイプが決定される。
  - ii. **手動入力による直接設定** プレイヤーが任意のMBTIタイプを直接指定することも可能。自動推定と併用して、後から修正することもできる。
- 設定されたMBTIは、イベントの発生傾向や会話内容、性格パラメータとの整合性などに間接的に影響する。

#### 3.1.3 性格パラメータ（5種）

MBTIとは別に、キャラクターには以下の5項目に関する**性格パラメータ**が存在する。各項目は1～5段階の整数値で構成される。これらは MBTI とは独立して設定されるが、合わせて性格全体を構成する。

パラメータ名	概要	ゲーム内での主な影響
社交性	他人との関係構築の積極性	雑談・挨拶イベントの発生頻度に影響
気配り傾向	他者への共感や配慮の度合い	感情ラベル変化や優しさ描写に影響
頑固さ	自己主張の強さ、意見の曲げにくさ	関係変化時の対立や意見交換描写に影響
行動力	行動の速さ・イベントの主導傾向	イベントの発端キャラとしての頻度に影響
表現力	感情や考えの言語的表出の豊かさ	会話の長さや表現の彩度に影響

- これらの値は、キャラ作成時にスライダーで設定することで決定される。
- 各パラメータ値は管理室にて後から編集可能。ただし、編集履歴や性格変化イベントによる変更は現仕様では非対応。

### 3.1.4 その他仕様との連携

- 性格プロフィールは、イベント発生確率やGPTによる会話生成（話し方や内容）に影響を与える重要パラメータである。
- これらの情報は、**キャラクター作成画面**および**管理室画面**で設定・編集・閲覧が可能。
- これらの情報はキャラステータス画面にも一部反映される（話し方、MBTI、性格パラメータ）。

**\*\*活動傾向（朝型・夜型）\*\***もイベント発生傾向に影響する要素のひとつであり、イベント抽選時の時間帯補正に利用される。

## 3.2 関係ラベルと初期関係

キャラクター同士の関係性は、**双方向同値の関係ラベル**によって管理される。これらのラベルは、特定のイベントまたは管理操作を通じて段階的に変化し、キャラクター間の親密度や関係の質を明確に表現するための指標である。

### 3.2.1 関係ラベルの定義と構造

関係ラベルは以下のような段階構造を持ち、キャラA⇄Bの間で同一のラベルが設定される：

なし → 認知 → 友達 → 親友 / 恋人 / 家族（排他的）

- 「なし」：初期状態。互いを認識していない関係。
- 「認知」：名前や姿を知っている状態。
- 「友達」：基本的な信頼関係が構築された状態。
- 「親友」「恋人」：高い信頼と特別な絆を意味する。
  - 「親友」は観察型イベントによって成立。
  - 「恋人」は**プレイヤー介入型イベント**（告白イベント）を通じてのみ成立。
- 「家族」：他のすべての関係ラベルと**排他的**であり、同時に保持できない特別な関係。初期設定または管理室での手動設定によりのみ付与される。

なお、「友達」からの関係ラベルの進展は、「親友」または「恋人」のいずれか一方に分岐する排他的な構造となっている。つまり、「友達」状態からは、観察型イベントによって「親友」へ、あるいはプレイヤー介入型の告白イベントによって「恋人」へと発展するが、両者は同時には成立しない。構造としては以下のように整理される：

↻ 親友  
友達  
↻ 恋人

一度「親友」または「恋人」となった場合でも、関係が破綻するなどして再び「友達」へ戻った後、別の分岐先へ進展する可能性がある。

### 3.2.2 関係ラベルの変化と発生イベント

関係ラベルは以下の遷移を通じて変化し、そのきっかけは**イベント発生**または**管理操作**による：

遷移	発生要因	プレイヤー関与
なし → 認知	挨拶イベント／時間経過	不要（自動）
認知 → 友達	「友達になるイベント（仮称）」	不要（観察型）
友達 → 親友	「親友になるイベント（仮称）」	不要（観察型）
友達 → 恋人	告白イベント	<b>必要（介入型）</b>
— → 家族	初期設定／管理室手動設定	任意（UI操作）

- \*\*「なし→認知」\*\*のみ時間経過やイベントでの自然変化が起こる。
- 「親友」は観察型イベントによってのみ到達し、プレイヤーの操作は不要。
- 「恋人」は**プレイヤーが選択肢などで関与することで成立**する唯一の関係ラベルである。
- 「家族」はイベントによって変化することはなく、明示的な設定によってのみ付与される。

### 3.2.3 関係ラベルの併存制限と優先順位

- 「親友」「恋人」「家族」は**相互に排他的**であり、いずれか一つのみを保持可能。
- 特に「家族」関係を設定した場合、他の関係ラベルは同時に保持できない。

- 既存のラベルが存在する場合、「家族」ラベルを設定すると**既存ラベルは上書きされる**（逆も同様）。

### 3.2.4 解除と別れイベントの予定

関係ラベルは一方向に進むのみではなく、将来的には「別れ（仮称）」イベントなどを通じて**解除・逆転が可能となる予定**である。

- 例：「親友」→「友達」への降格など
- プレイヤーが関与する選択肢型イベントや、長期末接触による強制変化なども検討中。

（現時点では未実装であり、将来拡張に備えたフックのみを用意）

### 3.2.5 初期関係設定機能

関係ラベルは、キャラクター追加時または管理室画面において初期設定することが可能である。また、関係設定時にキャラA→BおよびB→Aの\*\*呼び方（ニックネーム）\*\*をそれぞれ指定することもできる。これは家族関係に限らず任意の関係ラベルにおいて利用可能であり、会話生成や関係表示の際に使用される。

- **追加時**：既存キャラとの関係性として「なし」「認知」「友達」「親友」「恋人」「家族」から選択可。
- **管理室**：後から「家族」を含む初期関係を再設定することも可能。
- イベントによる変化とは無関係に、プレイヤーが任意に調整できる。

### 3.2.6 表示と通知仕様

- 関係ラベルは、キャラステータス画面およびキャラ関係詳細画面に表示される。
  - ステータス画面では一覧形式で簡潔に表示。
  - 詳細画面では、ラベルの変化履歴を含めた経過を確認可能。
- ラベルが変化した際には、ログ上に以下のようなSYSTEM行が表示される：

SYSTEM: AとBの関係が『恋人』になりました。

### 3.3 好感度仕様

好感度は、キャラクターAがキャラクターBに対してどれほど好意・親近感を抱いているかを表す数値であり、**A→Bの片方向**で定義される。関係ラベルや感情ラベルの変化にも間接的に影響を与える、シミュレーションにおける中核的な内部パラメータである。

#### 3.3.1 数値仕様

- **値の範囲**： `-100 ~ +100`
- **方向性**：原則としてプラス方向（上昇傾向）で推移するが、関係悪化イベント等によりマイナス域に至ることもある。
- **非対称性**：A→BとB→Aは**独立して管理**され、互いに影響しない（片思いや一方的な好意も表現可能）。

#### 3.3.2 初期好感度の設定方法

キャラクター追加時や管理室での初期関係設定の際に、**好感度も同時に設定可能**とする。これにより、すでに親しい（あるいは対立している）関係性を自然な数値で表現できる。

- プレイヤーは、A→B、B→Aの片方向ごとにスライダー形式で数値を設定する。
- スライダーの範囲は `-100~+100`、初期値は関係ラベルに応じて**自動的に提案される**。

例（UIイメージ）：

▼ 初期関係：恋人  
→ A→B 好感度：[ 50 ]    B→A 好感度：[ 50 ]  
（スライダーで調整可能）

関係ラベル	提案される初期好感度
なし	0
認知	5
友達	20
親友	40
恋人	50
家族	50

- プレイヤーはこの値を変更することで、非対称な関係（例：Aは好き、Bはそうでもない）を自由に設定可能。
- 初期設定後も、**管理室から好感度の再設定が可能**（ただし通常のプレイでは自然変化に任せることが推奨される）。

### 3.3.3 自然減少ルール（毎日評価）

好感度は、キャラクター同士の接触頻度に応じて**自動的に減少**することがある。これは、長期間にわたり接触がない場合、関係性が希薄化することを表現している。

最終接触からの経過日数	減少量
3日以内に接触あり	0
4～7日以内	-1
8日以上	-2

- \*\*「接触あり」\*\*とは、イベント（雑談・二人きり・思い出し など）で会話や交流が発生したことを指す。
- 評価は一日一回、自動的に処理される。

### 3.3.4 イベントによる変化

好感度はイベント発生により**固定値で増減**する。各イベントには、相手への好感度変化量があらかじめ定義されている。

- 例：
  - 雑談イベント発生：+1～+3
  - トラブルイベント：-3
- 数値は基本的に小さな変化の積み重ねを想定し、単一イベントで極端な変化が起きることは避ける。

---

### 3.3.5 好感度の利用先

好感度そのものはUI上で直接表示されるが、数値の大小は以下のように他要素に影響する：

用途	影響内容
感情ラベルの抽選確率	高いほど好意的なラベルへの変化確率が上昇
関係ラベル変化イベントの発生率	高いほど発生しやすくなる（特に「友達」→「親友」「恋人」）
特定イベントの台詞や反応	キャラの反応がより柔らかく／鋭くなること がある

### 3.3.6 UIでの表示

- **キャラステータス画面**にて、対象キャラに対する好感度が**バー形式で可視化**される。
- 数値は表示されないが、色の変化や長さによっておよその傾向が分かるようになっている。
- 好感度の相互関係（A→BとB→A）は**非対称**であり、それぞれ独立して管理される。

---

## 3.4 感情ラベル仕様

### 3.4.1 ラベルの種類と遷移構造



キャラクターA→Bに対して、以下のいずれか1種類の感情ラベルが付与される。ラベルは\*\*片方向性（A→Bのみ）\*\*で、B→Aとは無関係である。

#### 通常関係（関係ラベルが「友達」など）

嫌いかも ← なし → 気になる → 好きかも

- 「気になる」は**接触がない期間が続くと自然にリセット**される。
- 「好きかも」「嫌いかも」は自然には戻らず、**イベントによってのみ**変化する。

#### 特別関係（関係ラベルが「恋人」「親友」「家族」）

嫌い ← 普通 ← 好き（初期値） → 大好き

- 初期状態は「好き」。
- 「大好き」状態からは、特定のイベントでのみ「嫌い」や「普通」に変化。
- 「大嫌い」状態に遷移した場合、**「別れたい」などの解除イベント**が発生する。

#### 特殊ラベル「気まずい」

- あらゆる関係・感情ラベルから遷移可能。
- **時間経過により自然にリセット**される。
- 特定のイベントを通じて、他の感情ラベルへと戻る場合もある。

#### 備考

なお、「通常系列」と「特別系列（恋人・親友・家族）」は**明確に分離**されており、感情ラベルの変化は原則として**同一系列内でのみ**発生する（例：「好きかも」→「大好き」などの直接的な移行は発生しない）。特別系列への移行は、関係ラベルの変化により系列ごと切り替わることによってのみ起こる。

### 3.4.2 感情ラベルの変化タイミング

- 感情ラベルは以下のいずれかによって変化する：

- 非関与イベント時の抽選
- プレイヤー介入型イベント
- 時間経過（自然リセット）

感情ラベルは、原則としてイベントを契機とした抽選によってのみ変化する。ただし、以下の2つのラベルは例外的に\*\*時間経過による自然変化（リセット）\*\*が発生する：

- 「気になる」：未接触状態が7日間継続すると、自動的に「なし」に戻る。
- 「気まずい」：通常系列・特別系列を問わず、未接触状態が7日間継続すると、自動的にリセットされる。

※ この「接触」は、雑談・思い出し会話・二人きりの時間イベントなど、ログに記録される会話系イベントが対象となる。これらの変化は**抽選を伴わない**定常的な処理であり、ユーザーの操作有無にかかわらず進行する。

上記以外の感情ラベルについては、**時間による変化は一切発生しない**。

### 3.4.3 感情ラベルのログ出力

感情ラベルが変化した際は、**キャラクター同士の会話ログとは別に**、以下の形式でシステム出力される：

SYSTEM: キャラA→キャラBの印象が「気になる」に変化しました。

※ 「ラベル」という語は使用せず、「印象」に言い換えて自然な文体とする。

### 3.4.4 感情ラベルの変化抽選仕様

感情ラベルの変化は、イベントの発生時にのみ抽選される。抽選の有無や変化先は、以下の要素に基づいて決定される：

- イベント種別（雑談、思い出し会話、二人きりの時間など）
- 対象キャラクター間の関係ラベル（友達、恋人、親友、家族 など）
- 現在の感情ラベル、およびそこから遷移可能な変化先ラベル
- キャラクターA→B間の好感度（-100 ～ +100）

抽選は、各変化先ごとに定義された「好感度=20」「50」「80」の3点における代表的な変化確率（%）をもとに、好感度に応じて**線形補間**を行うことで、任意の好感度値における確率が算出される。

#### 変化確率の補間処理の例

```
def 感情変化確率(好感度: float, P20: float, P50: float, P80: float) ->
    if 好感度 <= 20:
        return P20
    elif 好感度 <= 50:
        return P20 + (好感度 - 20) * (P50 - P20) / 30
    elif 好感度 <= 80:
        return P50 + (好感度 - 50) * (P80 - P50) / 30
    else:
        return P80
```

#### 抽選の処理単位

- イベント1回につき、定義されたすべての変化先ラベルに対して**個別に抽選**が行われる（同時当選は不可。発生優先度は別途定義）。
- 全体として、変化が起こる確率はおおよそ**30%前後**となるよう調整されている。
- 「通常系列（なし～好きかも等）」および「特別系列（普通～大嫌い等）」のいずれも、上記と同様の補間ロジックを使用する。

#### 3.4.5 感情ラベルと関係ラベルの連動仕様（暫定案）

感情ラベルの変化により、関係ラベル変化イベントが発生する可能性がある（例：「好きかも」状態での会話により告白イベントが発生し、関係ラベルが「恋人」になる）。ただし、ラベル変化そのものは自動で起こらず、関係ラベルの遷移には**イベントを介したプレイヤー介入**が必須である。

今後のイベント設計において、感情ラベルをトリガーとする関係変化条件が設定される見込みである。

#### 3.4.6 初期設定時の自動付与ロジック

感情ラベルは、キャラクター追加時に設定される関係ラベルおよび好感度に  
応じて、以下のルールに従い自動的に初期値が決定される。この処理は片方  
向（A→B）ごとに実施され、プレイヤーが手動で編集することはできない。

✔ 通常関係（「認知」「友達」）

関係ラベル	初期好感度	感情ラベル（A→B）
認知	-50未満	嫌いかも
認知	-49～10	なし
認知	11以上	気になる
友達	-50未満	嫌いかも
友達	-49～15	なし
友達	16～59	気になる
友達	60以上	好きかも

✔ 特別関係（「親友」「恋人」「家族」）

関係ラベル	初期好感度	感情ラベル（A→B）
親友／恋人／家族	-50未満	嫌い
親友／恋人／家族	-49～30	普通
親友／恋人／家族	31～69	好き
親友／恋人／家族	70以上	大好き

※ 感情ラベル「気まずい」「大嫌い」は初期設定では付与されず、イベ  
ントまたは時間経過により変化するものとする。

### 3.5 信頼度仕様

#### 3.5.1 概要

信頼度は、キャラクターがプレイヤーに対して抱く信頼の度合いを数値で表したパラメータであり、\*\*一方向の値（キャラ→プレイヤー）\*\*として扱う。キャラクター同士の関係性とは無関係に、プレイヤーとの個別のつながりを表すため、感情ラベルや好感度とは別に管理される。

信頼度は、主に\*\*プレイヤー介入イベント（困りごと相談）\*\*におけるプレイヤーの対応内容によって上下する。

3.5.2 数値仕様

- **範囲：** 0 ～ 100
- **初期値：**キャラクターごとに異なり、以下のような**性格パラメータ**に応じて**設定されるランダム値**となる：

特性の例	初期信頼度の傾向値
警戒心が強い（内向的／頑固）	30～40
社交的 or 行動力が高い	60～70
通常（平均的）	50（中央値）

このように、初期信頼度は一律ではなく、キャラクターの性格に基づいた**幅を持たせた設定**となる。これにより、ゲーム開始時点からキャラクターごとの個性や距離感の差異が生まれるように設計されている。

3.5.3 変化の契機

信頼度の変化は、原則として**プレイヤー介入イベント**で発生する。具体的には以下のようなケースがある：

契機	変化方向	変化量の目安
プレイヤーの適切な対応（相談への好回答）	上昇	+3～+5
プレイヤーの曖昧・回避的な回答	上昇	+0～+2
プレイヤーの冷淡・否定的な対応	下降	-2～-4

なお、キャラクターの性格（例：気配り度が高い、表現力が高い など）により変化量が補正される場合がある。

### 3.5.4 表示と通知

信頼度の変化は、イベント終了時に**ログ上で簡易に通知**される：

SYSTEM: ○○からの信頼度が上昇しました。

この際、具体的な数値（+3など）は即座には表示されず、プレイヤーが後からステータス画面で確認できるようにする。

### 3.5.5 信頼度による影響

信頼度は、キャラクターがプレイヤーに対してどのように振る舞うかに影響を与える指標であり、ゲーム内のいくつかの要素に連動する。現時点では、主に以下の3つの側面に影響を及ぼす：

- **イベント発生頻度の補正** 信頼度が高いキャラクターは、プレイヤーに対してより頻繁に相談イベントを発生させるようになる。逆に、信頼度が極端に低い場合は、イベントの出現頻度が減少する（→3.5.6）。
- **相談イベントの内容の深さ** 信頼度に応じて、より打ち明けがいのある深い内容の相談が解放されていく。特にLv2以降の感情的・内省的な話題は、一定以上の信頼が構築されていないと発生しない（→3.5.6）。
- **プレイヤーへの態度や話し方の変化** 信頼度が高まると、キャラクターの口調や語彙に打ち解けた雰囲気や親しみがにじむようになり、逆に低い場合はよそよそしさや警戒心が表現される（→3.5.7）。

### 3.5.6 信頼度による相談イベント出現と内容の深さ

信頼度は、**相談イベントの発生確率**と、\*\*イベントの内容の深さ（レベル）\*\*に関係する。

#### (1) イベント発生頻度の補正

相談イベントは、信頼度に応じて発生確率が変化する。補正には以下の計算式が用いられる：

発生率補正係数 = 0.5 + (信頼度 ÷ 100) × 補正幅

- 補正幅の初期設定値は 0.5 とする。
  - 例：信頼度が 100 → 補正係数 1.5倍
  - 例：信頼度が 50 → 補正係数 1.0倍（基準値）
  - 例：信頼度が 0（理論上の最小値）→ 補正係数 0.5倍

この補正係数はイベント抽選時に乗算され、出現確率を増減させる。

## （2）相談内容の深さと解放条件

相談テンプレートは、信頼度に応じて「Lv0～Lv3」の段階を持つ。これらは話題の重さ・内面の深さを示しており、高信頼であるほど深い内容が発生可能となる。

信頼度	出現候補となるレベル	出現比率の傾向（例）
0～20	Lv0 のみ	100% Lv0
40	Lv0～Lv2	約60% Lv0 / 30% Lv1 / 10% Lv2
60	Lv0～Lv2	約45% Lv0 / 40% Lv1 / 15% Lv2
80	Lv0～Lv3	約25% Lv0 / 45% Lv1 / 25% Lv2 / 5% Lv3
100	Lv0～Lv3	約15% Lv0 / 40% Lv1 / 30% Lv2 / 15% Lv3

※ 比率の具体値は 5.3.3 内容の選択と形式 にて定義される。

## （3）低信頼度専用テンプレート

信頼度が 20以下 の場合、通常の相談テンプレートは使用されず、**低信頼度専用テンプレート**が選出される。

- このテンプレート群では、キャラクターの性格は維持されるが、プレイヤーに対して**警戒・戸惑い・皮肉**などがにじむように調整されている。

- トーンや内容の方針は 5.3.3 にて詳細定義されている。

### 3.5.7 プレイヤーに対する態度変化（tone差分）

信頼度の段階は、キャラクターのプレイヤーに対する態度や話し方に影響する。これにより、同じキャラクターでも信頼度に応じて印象が大きく変化するよう設計されている。

#### （1）信頼度段階と対応するトーン傾向

信頼度範囲	段階名	雰囲気・話し方の傾向
0～20	警戒	よそよそしい、皮肉、冷たい
21～50	距離あり	礼儀的、敬語ベース、探るような口調
51～80	好意的	自然体、やや親しみのある話し方
81～100	親密	打ち明け、甘え、感謝、やわらかな口調など

この分類は、ログ演出やセリフの印象づけに使われるだけでなく、GPT出力用のtone指示にも連動する。

#### （2）GPTプロンプトにおけるtoneタグ

プレイヤーに対してキャラクターが発するセリフは、信頼度に応じて以下のようなタグを付与して生成される：

```
{
  "to_player": {
    "trust_level": 87,
    "recognition_stage": "trust",
    "tone_tag": "trusting_friendly",
    "style_modifiers": ["emotive"]
  }
}
```

- `recognition_stage`：段階名（例：guarded, neutral, friendly, trust）



- `tone_tag` : 会話の雰囲気（例：guarded, formal\_neutral, friendly\_natural, trusting\_friendly）
- `style_modifiers` : 表現力などの性格パラメータによる補正

これらのラベルは 5.4 GPT出力テンプレート構造 にて、プロンプト構成とあわせて定義される。

---

## 3.6 ステータス画面構成

### 3.6.1 概要

ステータス画面は、各キャラクターの情報をプレイヤーが個別に確認するためのUIであり、「性格」「関係性」「信頼度」など、複数の要素をまとめて表示する場として機能する。情報は主に以下の3カテゴリに分けて提示される：

- プレイヤーとの関係
- 他キャラクターとの関係
- キャラクターの性格・傾向

本画面は主に情報確認を目的としており、編集機能は原則持たない。

### 3.6.2 表示内容

#### ■ 基本情報セクション

- キャラクター名
- 性格傾向
  - MBTI（推定タイプ）
  - スライダー形式による性格5項目（社交性／気配り傾向／頑固さ／行動力／表現力）

#### ■ プレイヤーとの関係情報

- 信頼度（バー表示、デフォルトでは数値非表示）

※ キャラ→プレイヤー間には感情ラベルは存在しない。

#### ■ 他キャラとの関係一覧

- 一覧形式で全キャラクターとの関係ラベルを表示
  - 例：「〇〇：友達」「××：恋人」
- 各関係ラベルはクリックで詳細ポップアップ／遷移可能

#### ■ 好感度・感情ラベル（キャラ⇄他キャラ）

- **好感度**：他キャラクターごとにバー表示（片方向ずつ）
- **感情ラベル**：一人につき一件（キャラA→キャラB）、好感度に加えて印象の傾向を補足する要素
- 感情ラベルの表示は好感度バーの補足情報として併記される（例：バーの横に「好きかも」など）

※ 数値は画面上では表示せず、ビジュアルのみで相対的に確認可能

#### ■ 簡易イベント履歴（抜粋）

- 直近数件のイベントログを簡略表示（例：「〇〇と雑談した」など）
- 詳細は日報画面と連携して閲覧可能

### 3.6.3 UI構成と操作

- セクションごとにタブまたは展開形式で構成
- 他キャラとの関係一覧は折りたたみ可能

なお、好感度・信頼度のバーは数値非表示とし、あくまでビジュアルによって傾向を伝える構成とする。

ユーザーにとっては「関係の空気感」を重視する体験とするため、数値的な管理や調整は基本的に意識されない設計である。

開発およびデバッグ用には、特定の操作（開発モードや検証用UI）により数値を確認可能とする機能の追加を検討中であるが、通常のプレイ画面では表示されない。

### 3.6.4 備考・補足

- 感情ラベルに絵文字や色分けなどの視覚的装飾は行わない

- 関係ラベルは一人のキャラが複数所持することはない（「家族」も併存不可）
  - 初期関係の設定結果もここに反映され、一覧上で確認可能
-

## 4. イベント仕様

### 4.1 イベントの基本構造

#### 4.1.1 イベントの分類

本作におけるイベントは、大きく以下の2種類に分類される：

分類	プレイヤーの関与	主な役割	例
非関与型 (観察型)	なし	キャラクター間の関係進展の観察	雑談イベント、思い出し会話、初対面、挨拶など
関与型	あり	プレイヤーの選択で結果が分岐	困りごと相談（プレイヤー介入イベント）

補足：「関係ラベル」の変化を引き起こすイベントは主に観察型イベントだが、「恋人」への関係変化はプレイヤーの関与が必要となる。

#### 4.1.2 イベントの基本構成要素

すべてのイベントは、以下の基本構造を持つ：

##### 1. 導入部

- イベントの趣旨が簡潔に説明される。例：「〇〇と××が会話中...」

##### 2. 本体（会話・行動の描写）

- GPTにより生成されたキャラクター同士の自然な会話
- 性格・関係性・時間帯などに基づいて内容が変化

##### 3. 変化の適用（オプション）

- イベントの種類・内容に応じて、好感度・信頼度・関係ラベル・感情ラベルが変化する

## 4. ログ出力

- プレイヤーに対し、発生イベントの概要と変化内容を提示（詳細は4.4 を参照）

### 4.1.3 イベント発生後の影響

イベントによっては、以下のような効果を伴う：

- **好感度の変化**
  - 多くのイベントで基本的に好感度は上昇
- **感情ラベルの抽選**
  - イベント内容と好感度に応じ、感情ラベルが変化する可能性がある（変化確率は約30%）
- **関係ラベルの変化**
  - 指定のイベントを通じて段階的に変化（例：「認知→友達」など）
- **信頼度の変化**
  - プレイヤー介入イベントのみで変動する（±固定値、数値は非表示）

なお、変化が起こった場合はすべて **SYSTEM:** ログに記載される。

---

## 4.2 イベントの発生タイミングと条件

### 4.2.1 ゲーム内時間と時間帯の扱い

本作では、ゲーム内時間は現実時間と同期して進行する仕様である。プレイヤーがアプリを開いた時点の「時間帯」によって、発生するイベントの種類や内容に偏りが出る。

時間帯分類（暫定）

分類名	対応する現実時間（例）	備考
朝	6:00～10:59	登校・通勤前の時間帯
昼	11:00～15:59	学校・仕事中の時間帯
夕方	16:00～18:59	放課後・帰宅時間帯
夜	19:00～23:59	自由時間・交流が活発になる
深夜	0:00～5:59	活動量が少ない

※ 実際の分類基準や発生率の調整は今後検討される可能性がある。

#### 4.2.2 キャラクター状態による制限

イベントは、キャラクターの状態によって発生可否が制限される。以下は主な条件の一例：

状態	発生可能なイベントの例	備考
通常	全てのイベントが発生対象	
就寝中	非発生状態	すべてのイベントが対象外
風邪	雑談・挨拶など一部軽度なイベントのみ	イベント頻度も減少傾向

キャラごとの活動傾向（夜型・朝型など）に応じて、特定時間帯に起こりやすいイベントを設定することも可能である。

#### 4.2.3 キャラクター性格による発生確率の傾斜

キャラクターの性格により、イベント種別ごとの発生しやすさが変化する。

性格要素	発生しやすいイベント
社交性が高い	雑談イベント、初対面イベント
社交性が低い	思い出し会話イベント、二人きりの時間イベント
行動力が高い	イベントの発端側（自分から話しかける等）
気配りが高い	プレイヤーへの相談イベント

※ 本発生傾斜は確率的なものであり、必ずしも性格と一致するとは限らない。

#### 4.2.4 イベント進行とプレイヤーの不在

イベントは、プレイヤーがアプリを開いていない間にも内部的に進行する。ただし、非関与型イベントの発生ログはプレイヤーがアプリを開いた時点でまとめて表示される（例：「〇〇と××が会話中...」）。

なお、プレイヤーが不在（アプリを開いていない）だった時間帯に関しても、内部では定期的にイベント抽選処理が行われていたように見せる演出を採用する。

具体的には、アプリを再び開いた際に、その時間帯に発生したとみなされるイベントをまとめてログ出力し、現実時間に追従してキャラクター同士の関係性が進行しているように感じられる構成となっている。

このとき、抽選処理自体はプレイヤーの操作とは独立して非同期で行われ、ログには自然な時間スタンプが付与されて表示される。これにより、アプリを継続的に起動していなくても、キャラクターたちの関係性が時間とともに進んでいるように見える。

---

### 4.3 イベントの種類一覧と概要

本作では、イベントはその性質や役割に応じて複数のカテゴリに分類される。以下に主要なイベント種別と、それぞれの概要を示す。

#### 4.3.1 雑談イベント

- **分類：**非関与型（観察型）

- **発生条件：**

- 関係ラベルが「友達」「親友」「恋人」「家族」のいずれかである
- 通常状態で活動中のキャラクター同士

- **内容：**

- その場の話題や季節、時間帯に応じた雑談を展開
- 会話後に好感度の上昇、および感情ラベルの抽選が行われることがある

#### 4.3.2 思い出し会話イベント

- **分類：**非関与型（観察型）

- **発生条件：**

- 関係ラベルが「友達」「親友」「恋人」「家族」のいずれか

- **内容：**

- 過去の出来事や以前の会話を思い出して再び話題にするイベント
- キャラクター同士の関係が深いときに発生しやすく、好感度上昇の余地が大きい

#### 4.3.3 二人きりの時間イベント

- **分類：**非関与型（観察型）

- **発生条件：**

- 状況的に二人きりになれる場面（他キャラクターが不在など）
- 関係ラベルや感情ラベルに関係なく発生可能

- **内容：**

- 通常より親密な雰囲気での会話が展開される
- 感情ラベル変化の抽選確率がやや高く設定されている

#### 4.3.4 挨拶イベント



- **分類：**非関与型（観察型）
- **発生条件：**
  - 関係ラベルにかかわらず発生可能
  - 比較的接触頻度が少ないキャラクター同士に発生しやすい
- **内容：**
  - 挨拶を交わす短い会話イベント
  - 初対面を演出する場合もあり、関係ラベル「なし」→「認知」への移行契機にもなる

#### 4.3.5 プレイヤー介入イベント（困りごと相談）

- **分類：**関与型（選択式 or 穴埋め式）
- **発生条件：**
  - キャラクターの内部状態に応じて随時生成
  - 信頼度が一定以上あるキャラクターに優先発生
- **内容：**
  - キャラクターがプレイヤーに助言や相談を求めてくる
  - プレイヤーの対応により信頼度が増減する
  - イベント終了時にはキャラクターが短くリアクションを返す（例：「ありがとう」など）

#### 4.3.6 備考

- 以下のイベントは現在未実装であり、将来的に追加予定：
    - 「認知」→「友達」：友達になるイベント（非関与型）
    - 「友達」→「親友」：親友になるイベント（非関与型）
    - 「友達」→「恋人」：告白イベント（プレイヤー介入型）
    - 「親友」or「恋人」→「友達」：別れイベント（介入型予定）
-

## 4.4 ログ出力ルールと形式

本作では、発生したイベントに関する情報がリアルタイムで画面内にログ表示される。ログの主な目的は、**キャラクター間の出来事を観察しやすくすること、および関係変化や感情の変動を視覚的に確認可能にすること**である。

### 4.4.1 ログの構成

ログは以下の要素で構成される：

種別	表記例	備考
イベント発生	EVENT: ○○と△△が会話中...	会話や行動イベントの導入部分。発生時点で1行出力。
システム通知	SYSTEM: ○○→△△の印象が「好きかも」に変化しました。	感情ラベルや関係ラベル、好感度・信頼度の変化等。
空行		イベントの切れ目に1行空けて視認性を確保する。

### 4.4.2 出力ルールの詳細

#### (1) イベントログ (EVENT)

- 会話・行動イベントが発生した時に、冒頭に1行だけ表示される。
- 表記は「○○と△△が○○中...」「○○がプレイヤーに相談しています...」などの定型。
- 会話の内容そのものはログには表示されない（非公開）。

#### (2) システムログ (SYSTEM)

- イベント終了時点で変化があった場合にのみ表示される。
- 以下のような形式で記載：

SYSTEM: ○○→△△の印象が「好きかも」に変化しました。  
SYSTEM: ○○→△△との関係が「親友」になりました。  
SYSTEM: ○○からの信頼度が上昇しました。

- 好感度や信頼度の変化については、数値ではなく「上昇」「下降」と表記。
- 一つのイベント内で複数の変化があった場合は、ログ内に複数行表示。

### (3) 表示位置・アニメーション

- ログは画面下部（CLI風ログ表示エリア）に表示。
- 新しいログは下から上へ積み重なる形式で表示。
- ログの更新はアニメーションで表示され、内容の自然な流れを演出。

#### 4.4.3 特記事項

- ログはゲーム内でのプレイヤーに対する主要なフィードバック手段であり、**キャラクターの独白や内心は表示されない**。
  - プレイヤー介入イベントでは、**プレイヤーの選択内容そのものはログには残らず**、あくまでその結果のみが表示される。
  - イベント履歴の保存に関しては別途日報画面にて確認可能。
-

## 5.内部処理・抽選仕様

---

### 5.1 イベント発生抽選

この節では、キャラクター間で自動的に発生する非介入型イベント（雑談・挨拶・思い出し・二人きり等）の抽選処理について記述する。

#### 5.1.1 候補ペアの生成ルール

非介入型イベントは、対象となる**キャラクターのペア**をあらかじめ生成し、その上でイベント種別を選定し、最終的に抽選・実行される。

対象ペアの生成条件：

- **共通条件（全イベント種別共通）**
  - 両者ともに状態が「就寝中」「風邪」等の制限状態でないこと
  - 両者ともに**初期設定済みであり、削除フラグが立っていないこと**
- **種別ごとの補正条件**
  - 「雑談イベント」：関係ラベルが「友達」「親友」「恋人」「家族」
  - 「挨拶イベント」：関係ラベルが「なし」「認知」「友達」「親友」「恋人」「家族」
  - 「思い出し会話イベント」：関係ラベルが「友達」「親友」「恋人」「家族」
  - 「二人きりの時間イベント」：関係ラベルが「友達」「親友」「恋人」「家族」

ペアの生成方法（例）：

全キャラクターの中から、条件を満たす順序付きペア（A→B）を列挙。ペア数が多すぎる場合は、一定数（例：上限20ペア）までに絞り込み。

#### 5.1.2 イベント種別の選定ロジック

対象ペアごとに発生可能なイベント種別を抽出し、その中から一つを選定する。

優先度設定（例）：

イベント種別	優先度	備考
挨拶	高	接触の起点となるイベント
雑談	中	ランダム性ありつつ頻出
二人きり	低	
思い出し会話	低	

※ただし、優先度は「発生率」ではなく「候補として選ばれる順序」を意味する。

### 5.1.3 発生確率の調整要因

最終的なイベント抽選において、以下の要素が発生確率に影響する：

- キャラ同士の関係ラベル（親しさが高いほど会話イベントが発生しやすい）
- 好感度（片方向が極端に低いと発生率が下がる）
- 性格パラメータ（社交性が高いキャラほど雑談が多く、行動力が高いキャラほど発端役になりやすい）
- 感情ラベルの状態（「嫌いかも」「気まずい」などを持っていると抽選の重みが大きく減衰する）
- 時間帯（朝や夜間など、キャラの活動傾向に応じて補正）

### 5.1.4 発生抑制条件と制限

以下のような状況では、イベント発生が**抑制または禁止**される：

- 同一のキャラが前回イベントから一定時間が経過していない（連続イベント抑制）
- 同一のペアで直前にイベントが発生していた
- 他の重要イベント（プレイヤー介入イベントなど）が画面上で処理中

### 5.1.5 実装設計と補足情報

非介入型イベントの抽選処理は、以下のような構成で実装される。

## 関数構成（擬似コード）

```
function runEventCycle() {  
  const pairs = generateCandidatePairs();  
  const candidates = evaluateEventCandidates(pairs);  
  const selected = drawFromCandidates(candidates);  
  if (selected) {  
    applyEvent(selected);  
  }  
}
```

## 補正要素と評価項目

各イベント候補には以下の情報が付与され、最終的な重みとして評価される：

- `baseRate`：イベント種別ごとの基本優先度（例：雑談=5、思い出し会話=8）
- `modifiers`：補正係数。以下の要因を考慮する：
  - 好感度（低すぎる場合は減衰）
  - 感情ラベル（「嫌いかも」「気まずい」→重みが大きく減衰）
  - 性格（社交性・行動力）
  - 時間帯と活動傾向の一致

## 抽選処理

評価されたイベント候補群に対して重み付き乱数抽選を行い、1件を選出する。

```
function drawFromCandidates(candidates): Candidate | null {
  const weighted = candidates.map(c => ({
    ...c,
    weight: calculateFinalWeight(c.baseRate, c.modifiers)
  }));
  return weighted.length > 0 ? weightedRandomChoice(weighted) : null
}
```

## イベント発生処理

抽選されたイベントは以下の処理を伴って実行される：

- GPT出力によるセリフ生成（プロンプトは別仕様）
- 好感度の固定上昇（+1～+3など）
- 感情ラベルの変化抽選（別仕様）
- ログ出力（EVENT：／SYSTEM：）の記録

## その他補足

- 補正值や重みは定数管理で調整可能にしておく
- 将来的に「イベント雰囲気（ポジティブ／ネガティブ）」による分岐演出に対応可能
- 同一ペア連続イベントを避ける履歴保持処理なども実装可能

---

## 5.2 時間帯補正仕様

本節では、ゲーム内の時間帯（朝・昼・夜など）に応じたイベント発生補正ロジックについて記述する。時間帯はイベント抽選時に使用され、発生率やキャラの行動傾向に影響する。

### 5.2.1 時間帯の区分

ゲーム内の時間は24時間制で扱われ、以下の時間帯に分類される。

時間帯名	時刻範囲（例）
朝	05:00～10:59
昼	11:00～15:59
夕方	16:00～18:59
夜	19:00～23:59
深夜	00:00～04:59

※ 各キャラクターの活動傾向は、以下の時間帯分類に基づいて補正される。

### 5.2.2 ベース発生率補正（全体）

時間帯ごとの基本発生倍率（全キャラ共通）は以下のとおり：

時間帯	補正倍率
朝	0.7
昼	1.0
夕方	1.2
夜	1.1
深夜	0.5

この倍率は、イベント候補の基本発生重みに掛けて適用される。

### 5.2.3 キャラクター活動傾向の補正

各キャラには「活動傾向（activityPattern）」として、以下のいずれかが設定される：

- `normal`（標準型）
- `morning`（朝型）
- `night`（夜型）

これに応じて、時間帯ごとの補正倍率に追加でキャラごとの個別補正がかかる。



活動傾向 \ 時間帯	朝	昼	夕方	夜	深夜
normal	×1.0	×1.0	×1.0	×1.0	×1.0
morning	×1.3	×1.1	×1.0	×0.9	×0.7
night	×0.7	×0.9	×1.0	×1.1	×1.3

※ 補正は「全体倍率 × キャラ個別倍率」として掛け合わせて適用。

#### 5.2.4 実装形式（補正計算）

擬似コード：

```
function getFinalWeight(baseWeight, timeSlot, activityType) {
  const global = globalTimeModifiers[timeSlot];
  const personal = personalTimeModifiers[activityType][timeSlot];
  return baseWeight * global * personal;
}
```

補正係数は外部からJSONで管理可能：

```
{
  "globalTimeModifiers": {
    "morning": 0.7,
    "noon": 1.0,
    "evening": 1.2,
    "night": 1.1,
    "midnight": 0.5
  },
  "personalTimeModifiers": {
    "normal": {
      "morning": 1.0, "noon": 1.0, "evening": 1.0, "night": 1.0, "mi
    },
    "morning": {
      "morning": 1.3, "noon": 1.0, "evening": 1.0, "night": 0.7, "mi
    },
    "night": {
      "morning": 0.7, "noon": 1.0, "evening": 1.0, "night": 1.1, "mi
    }
  }
}
```

### 5.2.5 備考

- `activityPattern` はキャラクターデータ（6.2）に設定される。
- 補正はすべて「発生候補評価時（イベント種別ごとの重み算出時）」に使用される。
- 一部イベント（就寝・風邪・相談など）は時間帯補正の影響を受けずに発生する。

---

## 5.3 プレイヤー介入イベント（困りごと）の生成

この節では、画面内に一時的に表示される「困りごと」相談イベントの生成および管理に関する内部処理仕様を記述する。これらはプレイヤーが選択または入力に対応することで完了し、信頼度などの変化を引き起こす。

### 5.3.1 イベント生成タイミングと頻度

自動生成のタイミング：

- ゲーム内時間と同期し、**約1時間ごと**に生成処理を試行
- 画面に既に表示されている相談が**存在しない場合**に限り、新規生成される
- 生成処理は1回につき**1件のみ**

**同時発生制限：**

- 複数の相談が**同時に画面上に表示されることはない**
- 既存のイベントがプレイヤーに無視されている間は、次の生成は行われない

### 5.3.2 対象キャラクターの選定条件

困りごとの発信者となるキャラクターは、以下の条件をすべて満たす必要がある。

**必須条件：**

- 状態が「就寝中」「風邪」などでない
- 直近1時間以内に他の困りごとイベントを発生させていない

**補正条件：**

- **信頼度が高いほど発生確率が上昇**
- 外交性・表現力が高いキャラクターは選ばれやすい傾向

### 5.3.3 内容の選択と形式

プレイヤー介入イベントで表示される「困りごと」は、テンプレート化された相談文をもとに構成される。テンプレートは、形式と深さの2軸で分類される。

#### 1) テンプレートの分類と形式

テンプレートは以下の2形式に分類される：

種別	内容形式	プレイヤー操作
選択式	2～3択のうちから選ぶ	ボタンで回答
穴埋め式	一部が空欄の文章	キーボード入力で補完

形式はテンプレートごとに固定されており、イベント生成時に形式を混在させることはない。

## 2) 信頼度によるテンプレート出現制御

テンプレートは「深さ（レベル）」と呼ばれる分類を持ち、信頼度に応じて出現可能なテンプレートのレベル範囲が変化する。

### 1. テンプレートの深さ（Lv）と信頼度の関係

テンプレートには以下の4段階の深さがある：

レベル	内容の傾向
Lv0	表面的・日常的・雑談的な話題
Lv1	やや内面・感情に踏み込む話題
Lv2	感情的・自己に関わる話題
Lv3	打ち明け・将来・対人関係に関わる深い相談

出現するレベルは信頼度によって制御され、以下のように段階的に解放される：

信頼度	抽選候補となるレベル	出現比率（例）
0～20	Lv0 のみ	100% Lv0
+40	Lv0～Lv2	60% Lv0 / 30% Lv1 / 10% Lv2
+80	Lv0～Lv3	25% Lv0 / 45% Lv1 / 25% Lv2 / 5% Lv3
+100	Lv0～Lv3	15% Lv0 / 40% Lv1 / 30% Lv2 / 15% Lv3

テンプレートごとのレベル分類は別途データファイル内に定義されており、実際の抽選処理ではこの出現比率に基づいて候補が選ばれる。

## 2. 低信頼度専用テンプレートの出現

信頼度が 20 以下の場合、通常のテンプレートは出現せず、**低信頼度専用テンプレート**が選出対象となる。

- 使用されるテンプレートはジャンルA～Eに限定される（感情・打ち明け系は除外）
- 内容は雑談風を装いつつも、プレイヤーへの距離感・懐疑・戸惑いをにじませた構成
- 表現トーンは「丁寧だが理知的な皮肉」「観察者視点の問いかけ」「上位存在性への違和感」など

## 3. トーン指定との連携（補足）

テンプレートのレベルや信頼度は、生成されるセリフの雰囲気にも影響を及ぼす。セリフ生成時には、以下のような `tone_tag` が付与される：

```
{
  "to_player": {
    "trust_level": 87,
    "recognition_stage": "trust",
    "tone_tag": "trusting_friendly",
    "style_modifiers": ["emotive"]
  }
}
```

これにより、同一テンプレートでも信頼度の違いによって話し方や印象が変化するように制御される。

### 5.3.4 回答後の処理と自動消滅

プレイヤーが回答した場合：

- 該当キャラが「ありがとう！」などの返答を行い、信頼度変動（上昇 or 停止）を伴って終了

- 回答内容によって信頼度が変化するが、数値は**その場では表示されない**
- ログに「SYSTEM: ○○からの信頼度が上昇しました。」等を入力

プレイヤーが無視した場合：

- 表示から**約1時間経過後に自動消滅**
- 特別なログ出力は行われない

---

このように、プレイヤー介入イベントは**リアルタイム性と選択一回性を重視**した構成とし、UI上でも自然に表示・終了するように設計されている。

---

## 5.4 GPT出力テンプレート構造

本節では、プレイヤー介入イベントやその他の発話生成において、GPT API を用いたセリフ生成を行う際のテンプレート構造および制御パラメータの仕様を定義する。信頼度や性格パラメータに応じて、出力されるセリフのトーンや雰囲気を調整可能とすることを目的とする。

### 5.4.1 出力テンプレートの目的と概要

テンプレートに従ってセリフを生成する際、キャラクターごとに個性を保ちつつも、プレイヤーへの態度や現在の関係性に応じた出力調整が必要となる。そのため、以下の出力情報を付与し、GPTモデルにプロンプトとして渡す：

- `core_prompt` （テンプレートの本文）
- `tone_tag` （信頼度・態度に基づく話し方）
- `style_modifiers` （性格パラメータ等に応じた修飾指定）

### 5.4.2 出力用テンプレート構造（基本）

テンプレート1件に対し、GPT出力時に以下のような構造でプロンプトが生成される：

```
{
  "core_prompt": "今日はちょっとだけサボっちゃった。こういう日ってあるよね",
  "to_player": {
    "trust_level": 68,
    "recognition_stage": "friendly",
    "tone_tag": "friendly_natural",
    "style_modifiers": ["emotive", "gentle"]
  }
}
```

- `core_prompt` : テンプレート本文（穴埋め or 完全文）
- `trust_level` : 現在の信頼度数値（0～100）
- `recognition_stage` : 信頼段階ラベル（後述）
- `tone_tag` : トーン識別子（後述）
- `style_modifiers` : 修飾ワード（任意、複数可）

#### 5.4.3 信頼度との連動：tone\_tagの決定

信頼度に応じて、キャラクターがプレイヤーに対してどのような態度を取るかを制御する。以下は信頼度範囲と対応するtone\_tagの分類例である：

信頼度範囲	段階名	tone_tag例
0～20	guarded	"guarded"
21～50	neutral	"formal_neutral"
51～80	friendly	"friendly_natural"
81～100	trust	"trusting_friendly"

tone\_tag は出力全体のトーンを指示するため、GPTへのプロンプト構築時には system 指定やキャラ話法の説明内で明示的に使われる。

#### 5.4.4 性格パラメータとの連動：style\_modifiersの追加

キャラクターごとに設定されている性格パラメータ（例：表現力・気配り傾向など）に応じて、以下のような修飾ワードが `style_modifiers` として付与される。

パラメータ項目	高い場合に付与される例
表現力	"emotive", "expressive"
気配り傾向	"gentle", "considerate"
頑固さ	"firm", "convictional"
社交性	"open", "talkative"

これらの修飾語は複数付与可能であり、GPTに出力のニュアンスを指示する目的で利用される。

#### 5.4.5 フォーマット出力例と補足

以下に、信頼度と性格パラメータを含めた出力テンプレートの具体例を示す：

```
{
  "core_prompt": "今日はちょっとだけサボっちゃった。こういう日ってあるよね",
  "to_player": {
    "trust_level": 87,
    "recognition_stage": "trust",
    "tone_tag": "trusting_friendly",
    "style_modifiers": ["emotive", "expressive"]
  }
}
```

- `recognition_stage` は `trust_level` に応じて自動決定される。
- `style_modifiers` はキャラクターの性格値に応じて、0〜3個程度付与される。
- 使用しない項目（例：modifier不要な場合）は空配列ではなく省略してもよい。

この構造により、セリフテンプレートごとの多様性を保ちつつ、個別のキャラクター性や信頼関係の変化を自然に反映することが可能となる。

---

## 5.5 雰囲気の抽選処理



本節では、イベント発生時に同時に決定される「雰囲気 (mood)」に関する抽選仕様を記述する。雰囲気は感情ラベルの変化確率やセリフ傾向などに影響する、短期的な会話の空気・トーンを表す指標である。

### 5.5.1 雰囲気ラベル定義

雰囲気は整数値（-2～+2）で内部管理され、以下のように定義される。

値	英語ラベル	日本語ラベル	出力傾向
2	very_positive	とてもポジティブ	明るく親しみやすい、安心感
1	positive	ポジティブ	穏やかで前向き
0	neutral	普通	中立、自然体
-1	negative	ネガティブ	疲労感、違和感、よそよそしさ
-2	very_negative	とてもネガティブ	張り詰めた空気、不安、冷たさ

- 内部的には数値で保存・参照されるが、UIやログでは日本語表示に変換される
- セリフトーンや感情ラベル変化補正に影響を与える（詳細は5.5節）

### 5.5.2 雰囲気の抽選タイミングと処理概要

- 雰囲気はイベント発生時に1回だけ抽選される（イベント単位）
- 抽選結果はそのイベント中のみ有効（継続しない）
- 抽選に用いるパラメータは以下：
  - 関係ラベル (relation\_label)
  - 感情ラベル (emotion\_label) ※主にA→B視点
  - 好感度 (affection)
  - 信頼度 (trust) ※プレイヤー介入イベント時

### 5.5.3 雰囲気決定の処理フロー（概要）

1. キャラクターの関係性・感情ラベル・好感度などをもとに、候補となる雰囲気値（-2～+2）ごとの**出現比率テーブル**を取得
2. 好感度や感情ラベルに応じて補正処理を適用（例：好感度が高いと -2 が出にくくなる）
3. 重みに基づき1つの雰囲気値を確率的に抽選し、イベントに紐づける

#### 5.5.4 出現比率テーブル例

以下は「関係：友達、感情：気になる、好感度50」の組み合わせに対する一例：

```
{
  "-2": 5,
  "-1": 10,
  "0": 40,
  "1": 30,
  "2": 15
}
```

※ 実際の出現比率は、イベント種別やキャラクター性格等によってさらに補正される場合がある。

#### 5.5.5 プレイヤー介入イベントでの特例

プレイヤーが関与する「困りごと」イベントにおいては、雰囲気値の決定に **キャラ→プレイヤーの信頼度（trust）** が加味される。

- 信頼度が高い場合： 1, 0（ポジティブ～普通）に寄りやすい
- 信頼度が低い場合： -2, -1（とてもネガティブ～ネガティブ）に寄りやすい

#### 5.5.6 実装上の備考

- 雰囲気は mood: -2 ~ +2 の形式で保存される（英語ラベルは補助的）
- 出力・ログ表示・UIでは対応する日本語ラベルが表示される
- 雰囲気値は 5.5 感情ラベルの抽選 における変化率補正にも用いられる
- 雰囲気に応じたセリフトーン補正は今後のプロンプト調整にて反映予定

---

## 5.6 感情ラベルの抽選

本節では、イベント発生時における感情ラベルの変化抽選仕様について記述する。抽選結果により、キャラクターAからBへの感情ラベル（例：「気になる」「嫌いかも」など）が変化する可能性がある。

### 5.6.1 抽選対象

A→B 間でイベントが発生した場合、その直後に感情ラベルの変化抽選が実施される。

なお、「気になる」「嫌いかも」は減衰対象とする。抽選によって他のラベルに遷移する可能性を残しつつ、自然変化も併用する。

### 5.6.2 抽選処理の概要

1. 現在の感情ラベル、関係ラベルをもとに候補ラベルを取得
2. 5.4 で決定された雰囲気（mood）を取得し、候補ごとの重みを補正
3. 補正後の重みをもとに、確率的に1ラベルを抽選
4. 抽選結果が現在のラベルと異なる場合、ラベルを変更

### 5.6.3 重みテーブル構造

感情ラベル抽選に用いる重みテーブルは以下の形式で管理される：

```
{
  "relation_label": "friend",
  "emotion_label": "気になる",
  "table": {
    "positive_2": {
      "気になる": 50,
      "好きかも": 30,
      "なし": 20
    },
    "positive_1": { ... },
    "neutral": { ... },
    "negative_1": { ... },
    "negative_2": { ... }
  }
}
```

- positive\_2 などは雰囲気値 2 を意味する (very\_positive)
- 雰囲気値 (-2~+2) ごとに異なる重みが設定されている
- 重みの合計は100でなくてもよく、相対比で扱う

#### 5.6.4 雰囲気による補正の例

例：現在「気になる」で、雰囲気が -2（とてもネガティブ）の場合：

```
{
  "気になる": 60,
  "なし": 40
}
```

→ 前向きな変化（「好きかも」など）が除外され、後退方向のみが候補となる

一方、雰囲気が +2（とてもポジティブ）の場合：

```
{
  "気になる": 50,
  "好きかも": 30,
  "なし": 20
}
```

→ 上昇・後退の両方向が候補に含まれる

#### 5.6.5 実装備考

- 雰囲気に基づく重み補正は事前に定義されたテーブルに従う
- 抽選後にラベルが変化した場合は、ログ出力およびステータスに反映される

---

### 5.7 関係ラベルの変化

本節では、キャラクター間における「関係ラベル」の変化処理について記述する。関係ラベルはキャラA⇄Bの**双方向同値関係**であり、イベントを通じて段階的に変化する。

#### 5.7.1 概要

関係ラベルは、以下のような段階を持つ：

- なし
- 認知
- 友達
- 親友
- 恋人（※プレイヤー介入イベントでのみ到達可能）
- 家族（※初期設定時にのみ指定可能、他ラベルと併存不可）

多くの関係ラベルの変化は、**プレイヤーの介入を必要としないイベント（非介入イベント／観察イベント）**を通じて発生する。時間経過によって変化するのは「なし」→「認知」のみであり、それ以外の関係ラベルは、主にイベントによって変化する。なお、プレイヤーの介入を必要とするイベントを通

じて変化するのは、「友達」→「恋人」（および将来的に予定されている「恋人」→〇〇）など、一部のケースに限られる。

### 5.7.2 初期関係の設定

新規キャラクター追加時、あるいは後から「管理室」より、初期関係を手動で設定できる。「友達」「家族」「恋人」などのラベルを、すでに存在する他キャラに対して事前に付与しておくことができる。

なお、初期関係として「家族」を選択した場合は、他の関係ラベルと**併存できない**（排他的に扱われる）。

また、関係ラベルに応じて好感度の**初期値が自動的に変化**する仕様を採用しており、設定された関係に応じて+20～+70程度の好感度が初期値として与えられる。

### 5.7.3 関係ラベルの変化イベントと条件

現在の関係ラベル	イベント種別	イベント分類	プレイヤー介入	新しい関係ラベル
なし	挨拶イベント	観察型	不要	認知
認知	友達になるイベント（仮称）	観察型	不要	友達
友達	親友になるイベント（仮称）	観察型	不要	親友
友達	告白イベント	介入型	必要	恋人
（任意）	別れイベント（仮称）	介入型	必要	友達など

「家族」への変化イベントは存在せず、手動設定のみによる。

### 5.7.4 ラベル変化の記録と表示

関係ラベルの変化が発生した場合、ログ上に以下のような形で記録される：

SYSTEM: AとBの関係が「友達」になりました。

また、ステータス画面やキャラ関係詳細画面にて、最新の関係ラベルを確認可能。

### 5.7.5 注意点

- 関係ラベルは双方向で共通（ $A \rightleftharpoons B$ ）である。
- 一部ラベル（家族）は排他的で、他のラベルと併存できない。
- 一度「恋人」や「親友」になった場合でも、特定のイベントにより解除される可能性がある（例：別れイベント）。

---

## 5.8 ラベル変化の記録と表示

本節では、関係ラベルおよび感情ラベルが変化した際のログ記録形式と、画面上での表示仕様について記述する。

### 5.8.1 ログ出力の基本形式

ラベル変化が発生した場合、メイン画面に表示されるログには以下のような形式で記録される。

#### 関係ラベル変化時

SYSTEM: AとBの関係が「友達」になりました。

#### 感情ラベル変化時

SYSTEM: A→Bの印象が「気になる」に変化しました。

感情ラベルに関しては、双方向ではなく、片方向的な変化を示す表現を用いる。

### 5.8.2 出力タイミング

ラベル変化が発生したイベントの直後、その他のログ出力と合わせて即時に表示される。

イベントのログよりも下位に位置し、後続の会話ログが続く場合はその直後に差し込まれる。

### 5.8.3 出力対象の制御

以下の場合、ログは出力されない：

- 関係ラベル・感情ラベルに**変化がなかった**場合
- ラベルが「なし」にリセットされたが、**同一状態への遷移**である場合（例：既に「なし」で、リセット処理が行われたなど）

### 5.8.4 ラベルの表示位置（画面UI）

関係ラベル・感情ラベルの現在の状態は、以下の場所にて確認できる：

- **ステータス画面**（キャラ→他キャラとの関係一覧に表示）
- **キャラ関係詳細画面**（一対一の関係と変遷履歴）
- **管理室**（設定用の編集画面）

なお、ログ上では過去の変化履歴も一定期間保持され、ユーザーが任意に参照可能となる。

### 5.8.5 備考

- 表示文言に「ラベル」等の技術的表現は使用しない。
- 感情ラベル「気まずい」など、一時的な印象変化も記録対象とする。
- 表示に使用する文字列は統一し、システムメッセージとしてログ上に出る。

---

## 5.9 その他の変化仕様

本節では、ラベルや信頼度・好感度以外で、時間経過やイベントによって変化する要素について簡潔に記述する。

### 5.9.1 状態変化（例：「風邪」「就寝中」）



各キャラクターには一時的な「状態」が付与されることがあり、主に以下のような状態が存在する：

- **風邪**：体調が悪くなる。発言頻度や内容に変化が出る場合がある。
- **就寝中**：深夜帯などに自動的に付与。イベント発生抑制などに影響。

これらの状態は「みんなの様子」画面にて、キャラ名の横にテキストで表示される。

### 5.9.2 状態の付与と解除

- 状態は、**時間帯**や**イベントトリガー**により自動的に付与・解除される。
- 「風邪」状態は一定確率で発症し、数日間持続する。
- 「就寝中」はプレイヤーの端末時間に基づき、深夜0時～朝6時に自動的に付与される。
- その他の状態も、今後の拡張で追加予定。

### 5.9.3 UI上での反映

状態は以下の画面で視認可能：

- **みんなの様子（キャラ一覧）**
- **キャラステータス画面（補足情報エリア）**

状態の詳細な説明は基本的にUI上に表示されず、直感的な理解を重視する。

### 5.9.4 備考

- 状態変化はログとしては記録されない。
  - 状態が原因でイベントが発生しなかった場合も、特に通知や記録は行われない。
-

## 6. データ構造と保存仕様

---

本章では、アプリケーションにおける各種データの構造と、それらの保存および読み込みに関する仕様を記述する。UI設計や内部処理仕様と連動し、キャラクター、関係、イベント履歴等をどのように保持・管理・永続化するかを明確にすることを目的とする。

### 6.1 データ構造の全体概要

本アプリケーションでは、以下の主要な構成単位ごとにデータを管理する：

- キャラクター (Character)
- 関係 (Relationship)
- 感情ラベル (EmotionLabel)
- 状態 (Status)
- イベント履歴 (EventHistory)
- 日報 (DailyReport)
- プレイヤー情報 (PlayerData)

それぞれの構造や保存形式については以降の節にて詳述する。

---

### 6.2 キャラクターデータの構造

本節では、アプリケーション内でのキャラクター1人分のデータ構造について定義する。各キャラクターは固有のIDとともに、性格や話し方、状態、プレイヤーに対する信頼度などを保持する。

#### 6.2.1 JSON構造例

```
{
  "id": "char_001",
  "name": "碧",
  "mbti": "INFP",
  "mbti_slider": [3, 2, 1, 0, 2, 3, 1, 4, 2, 0, 1, 3],
  "personality": {
    "social": 4,
    "kindness": 3,
    "stubbornness": 2,
    "activity": 5,
    "expressiveness": 4
  },
  "talk_style": {
    "preset": "くだけた",
    "first_person": "俺",
    "suffix": "～じゃん"
  },
  "status": "風邪",
  "last_interacted": "2025-07-08T12:15:00Z",
  "trust": 45,
  "last_support": "2025-07-07T19:20:00Z",
  "activityPattern": "night"
}
```

### 6.2.2 各要素の説明

キー	説明
<code>id</code>	キャラクター固有のID。内部処理に用いる。
<code>name</code>	キャラクター名（表示用）
<code>mbti</code>	MBTIの4文字タイプ（例："INTJ"）
<code>mbti_slider</code>	MBTI診断に用いた12個のスライダー値（0～4）
<code>personality</code>	各種性格パラメータ（1～5）
<code>talk_style</code>	話し方のプリセットと一人称・語尾設定
<code>status</code>	現在の状態（例："活動中", "風邪", "就寝中"など）
<code>last_interacted</code>	最後に他キャラクターとイベントが発生した日時（ISO形式）
<code>trust</code>	プレイヤーに対する信頼度（0～100）
<code>last_support</code>	最後にプレイヤーの支援を受けた日時（ISO形式）
<code>activityPattern</code>	キャラの活動傾向（ <code>normal</code> / <code>morning</code> / <code>night</code> ） イベント抽選時の時間帯補正に使用

### 6.2.3 備考

- `mbti_slider` の順序は診断仕様に依存し、各項目が固定順で格納される。
- `status` の状態値は、ステータス画面やイベント抽選条件にも用いられる。
- `trust` はイベント発生頻度やセリフ内容に影響を与える可能性がある。
- `last_interacted` および `last_support` は好感度や信頼度の自然減少処理に活用される。
- `activityPattern` は「5.2 時間帯補正仕様」で定義される時間帯ごとの補正倍率に関与する。

## 6.3 関係・感情データの構造

本節では、キャラクター間の関係ラベル、感情ラベル、好感度、プレイヤーとの信頼度など、人間関係に関連するデータ構造について記述する。

6.3.1 関係ラベル (relation\_label)

- A↔B の双方向で1つのみ存在し、排他的なラベルである。
- 以下のような状態が存在する：

ラベル名	変化方法	備考
なし	初期状態	
認知	自動（挨拶イベント or 時間経過）	
友達	非介入イベント（友達になるイベント）	
親友	非介入イベント（親友になるイベント）	
恋人	プレイヤー介入イベント（告白イベント）	
家族	初期設定・管理室にて手動設定（固定）	他ラベルと併存不可、イベントによる変更不可

```
{
  "pair": ["char_001", "char_002"],
  "relation_label": "友達"
}
```

6.3.2 感情ラベル (emotion\_label)

- A→B の片方向で1つだけ持つ。
- 以下の系列が存在する：

「認知」「友達」以下の関係で使用される感情ラベル：

- なし（初期状態）
- 気になる
- 好きかも
- 嫌いかも
- 気まずい（特別な状態で、全関係ラベルで共有）

「親友」「恋人」「家族」の関係で使用される感情ラベル：

- 普通（初期状態）
- 好き
- 大好き
- 嫌い
- 大嫌い
- 気まずい

```
{
  "from": "char_001",
  "to": "char_002",
  "emotion_label": "気になる"
}
```

### 6.3.3 好感度（affection）

- A→B の片方向で管理。
- -100 ～ +100 の整数値で表現。
- 初期値は 0。ただし、初期関係の設定により値を変動させることが可能。

```
{
  "from": "char_001",
  "to": "char_002",
  "affection": 38
}
```

#### 6.3.4 信頼度 (trust)

- キャラクター → プレイヤー の片方向で管理。
- 0 ～ 100 の整数値で表現。
- 初期値は 0。

```
{  
  "character_id": "char_001",  
  "trust": 20  
}
```

#### 6.3.5 備考

- 関係ラベル (relation\_label) は常に双方向で共通である。
- 感情ラベル・好感度は片方向ごとに独立して管理され、変化も個別に処理される。
- 感情ラベルと関係ラベルの変化は、イベントや時間経過を通じて別途管理される。

#### 6.3.6 家族ラベルの特例仕様

- 「家族」ラベルは初期設定または管理室での**手動設定のみ**で付与される。
- イベントによる変化（付与・解除）は一切行われず、**完全に固定された関係**とみなされる。
- 他の関係ラベル（親友・恋人など）とは**\*\*併存不可（排他的）\*\***であり、設定時に他ラベルを上書きする。
- 一度「家族」ラベルを解除した場合は、再び他ラベルの設定が可能となる。

#### 補足情報の設定（種別）

- 「家族」には以下のような**種別情報**を付与可能：
  - 兄弟（兄／弟／姉／妹）
  - 親子（父／母／息子／娘）

- その他（祖父母、いとこなど）
- UI上では「家族（姉）」のように表示され、会話生成や態度に影響を与える。

#### 感情ラベルとの関係

- 「家族」関係においても、感情ラベルや好感度は**通常どおり管理・変化**される。
- 感情ラベルの系列は「親友」「恋人」と同一：
  - 嫌い ← 普通 ← 好き（初期） → 大好き（※「大嫌い」は使用不可）
- 大嫌い相当の状態になっても、**別れイベントは発生せず除外対象**となる。

---

## 6.4 ログ・履歴の保存仕様

本節では、アプリ内で生成される各種イベントログ、状態変化ログ、及びそれらの保存形式について説明する。

### 6.4.1 ログの種類

以下の種類のログが存在する：

ログ種別	内容	表示場所	保存対象
イベントログ	キャラクター同士の会話やイベント発生	メイン画面CLI風ログエリア	ログ履歴
システムログ	関係ラベル、感情ラベル、好感度等の変化通知	メイン画面CLI風ログエリア	ログ履歴
介入イベント応答	プレイヤーが選択した回答とその反応	メイン画面CLI風ログエリア	ログ履歴
日報ログ	各日終了時にまとめて出力されるイベント一覧	日報画面	日報履歴



### 6.4.2 ログの保存形式（CLIログ）

ログは日付ごとにまとめられ、以下のようなJSON形式で保存される：

```
{
  "2025-07-08": [
    {
      "time": "10:45",
      "type": "event",
      "content": "EVENT: 莉音と咲が会話中..."
    },
    {
      "time": "10:46",
      "type": "system",
      "content": "SYSTEM: 莉音→咲の印象が「気になる」に変化しました。"
    }
  ]
}
```

### 6.4.3 保存場所・保存方針

- ログはブラウザローカルストレージまたはIndexedDBに保存される想定。
- エクスポート・インポート機能によって任意の外部保存・読み込みが可能。
- 日報は別途、日単位で履歴として記録される。

### 6.4.4 日報の構造例

日報は以下のような簡潔な形式で記録される：

```
{
  "date": "2025-07-08",
  "events": [
    "10:45 莉音と咲が会話した",
    "15:22 志音と碧がすれ違った"
  ],
  "changes": [
    "莉音→咲：印象「なし」→「気になる」",
    "詠翔→響：関係「なし」→「友達」"
  ]
}
```

#### 6.4.5 備考

- ログはユーザーの設定により非表示にすることも可能だが、内部的には常に保存されている。
- 感情ラベルや関係ラベルの変化ログは、システムログとして自動出力される。
- 日報は CLI ログを元に、一定ルールに従って抽出・整形される。
- CLIログと日報ログは両方ともセーブデータに含まれる。

---

### 6.5 セーブ・ロード仕様

本節では、ゲーム内の状態を保存・復元するためのセーブおよびロード機能の設計仕様を記述する。

#### 6.5.1 セーブ対象のデータ

セーブ機能では、以下のデータが保存対象となる：

- キャラクターデータ（プロフィール、性格、信頼度など）
- 関係・感情・好感度データ
- 呼び方データ（nickname）
- イベント履歴（CLIログ）
- 日報履歴
- プレイヤーによる設定（名前やUI設定など）

- キャラ一覧の状態（体調、就寝中など）

### 6.5.2 セーブ形式

セーブデータは、JSONフォーマットで構造化される。全体は以下のような構成となる：

```
{
  "save_id": "slot1",
  "datetime": "2025-07-08T23:00:00Z",
  "player_settings": {
    "name": "管理人さん"
  },
  "characters": [...],
  "relationships": [...],
  "nicknames": [...],
  "logs": { ... },
  "reports": [...],
  "status": { ... }
}
```

### 6.5.3 ロード仕様

- ロード時には、保存時点の状態を完全に復元する。
- セーブデータ選択画面から任意のデータを選択可能。
- 復元後、画面上には「復元完了」の表示がされる。
- システム内部時間はロード時点では現在時刻と同期されるため、時間経過による差異が発生する。

### 6.5.4 セーブスロット

- セーブスロットは最大で5つまで保持可能。
- オートセーブは1スロットに固定され、他4つはマニュアルセーブ用。
- セーブスロットは日時と簡易説明文（任意）で管理される。

### 6.5.5 バックアップ機能

- 外部出力（ファイル形式）によるセーブデータのバックアップが可能。

- 形式： `.json`
- インポート時には現在のデータを上書きするか、別スロットに保存するかを選択可能。

#### 6.5.6 備考

- セーブ機能は基本的に内部ストレージ（ブラウザのローカルストレージやIndexedDB）を使用。
- 外部バックアップはユーザーの任意操作による。
- 保存形式は今後のアップデートでも拡張可能な構造を意識して設計されている。

---

## 6.6 呼称データの構造

本節では、キャラクター間で使用される「呼び方」に関するデータ構造を定義する。呼称データは片方向で設定され、表示名や会話ログ中の代名詞として用いられる。

### 6.6.1 構造と仕様

- 各キャラクターは、他キャラクターに対して任意の呼称を設定できる。
- 呼称は **A→B** の片方向で管理される（B→Aとは無関係）。
- 呼称が未設定の場合は、デフォルトの名前（ `name` ）が使用される。
- UI上では「キャラ編集」や「情報確認画面」で表示・編集可能。
- 初期設定時や関係ラベル「家族」の設定時にも併せて入力できる。

### 6.6.2 設定方法

- 設定箇所：管理室 → キャラ追加 or 編集 → 初期関係設定
- 入力項目：
  - A→Bの呼び方（テキスト自由記入）
  - B→Aの呼び方（オプション、空欄可）

### 6.6.3 JSON構造例

```
{
  "from": "char_001",
  "to": "char_002",
  "nickname": "にいちゃん"
}
```

#### 6.6.4 備考

- 一人のキャラクターが複数人に対して異なる呼び方を設定することが可能。
- 同一の `from` と `to` のペアは1件のみ保持され、上書きで更新される。
- 呼称はセリフ生成時やステータス表示、CLIログなどに反映される。
- 絵文字や特殊文字も使用可能だが、長すぎる呼称はUI上で省略表示される場合がある。
- 片方向のみ設定された場合でも正常に表示され、相手側には影響しない。
- 家族関係の補足情報（兄弟・親子など）と連携して、自然な呼び方になるよう使用される。

---

### 6.7 テンプレート構造と保存仕様

本節では、相談イベント等で使用されるテンプレートデータの構造、および保存・抽出に関する仕様を定義する。テンプレートはキャラクターごとのデータとは独立しており、共通のイベント資源として全キャラクターで共有される。

#### 6.7.1 テンプレートデータの構成概要

テンプレートは「困りごと相談イベント」の表示内容を構成する文章群であり、以下の特徴を持つ：

- キャラクター個別ではなく、全体の共通データとして保持される
- 各テンプレートは、ジャンル・深さレベル・形式といった分類情報を持つ

- 出力時には信頼度や性格パラメータに応じて、抽選・フィルタを経て選出される（→5.3.3、5.4）

保存形式はJSONで統一され、各テンプレートは一意のIDを持つ。テンプレート一覧はファイル単位で分割されていてもよく、読み込み時に結合される。

### 6.7.2 テンプレート項目の定義

テンプレートオブジェクトは、以下の要素を持つ：

項目名	型	説明
id	string	テンプレートID。ファイル内一意。例："A012"
genre	string	テンプレートのジャンル分類（A～E）
level	int	深さレベル（0～3）
form	string	形式："choice" または "fill"
core_prompt	string	出力用セリフの本文（プレーンテキスト）

ジャンルやレベルの意味付け、使用範囲の制限については「5.3.3」にて詳述されている。

### 6.7.3 JSON構造例と解説

以下に、テンプレートデータの例を示す：

```
{
  "id": "B027",
  "genre": "B",
  "level": 1,
  "form": "fill",
  "core_prompt": "今日はちょっとだけサボっちゃった。こういう日ってあるよね"
}
```

このデータは以下のように解釈される：

- ジャンルB（例：日常／悩み）のテンプレート

- 信頼度が中程度（目安：40以上）で出現可能なLv1
- プレイヤーが空欄を埋めるタイプの穴埋め式
- 出力時にはこの本文が `core_prompt` としてそのまま渡される（→5.4）

#### 6.7.4 保存・取得に関する補足

- テンプレートは複数ファイルに分割しても構わず、読み込み時に統合管理される
- 出力時には、信頼度・キャラ性格・テンプレレベルなどをもとに**条件抽出 + 重み抽選**が行われる（→5.3.3.2）
- 将来的に `tone_tag` などをテンプレ側に記述する可能性もあるが、現時点ではテンプレート側では保持しない
- 抽選後に補助データ（`tone_tag`等）を付加して `core_prompt` と一緒に出力構造を構成する（→5.4）

---

## 7. MBTI診断仕様と性格タイプ決定

本章では、キャラクター性格タイプ（MBTI）を診断・設定するための仕様を定義する。診断はスライダー形式の16問によって行われ、プレイヤーは全キャラ共通の診断画面で初期性格を設定できる。

### 7.1 MBTI診断方式とロジック

MBTIは以下の4軸（8タイプ）から成り立つ：

軸名	左	右
E-I	外向（Extraversion）	内向（Introversion）
S-N	現実（Sensing）	直感（iNtuition）
T-F	思考（Thinking）	感情（Feeling）
J-P	判断（Judging）	知覚（Perceiving）

#### 7.1.1 診断質問と構成

診断には、以下の16問のスライダー質問を使用する。各軸に異なる観点で2問ずつ設けられている。



質問番号	軸	内容（左 = 前者、右 = 後者）
Q1	E-I	初対面の人ともすぐ話せるタイプ？ ←→ 少し様子を見てから話すタイプ？
Q2	S-N	話すときは具体的な話が多い？ ←→ 抽象的な話が好き？
Q3	T-F	決断は論理で考える？ ←→ 気持ちを重視する？
Q4	J-P	計画通りに物事を進めたいタイプ？ ←→ 成り行きに任せたいタイプ？
Q5	E-I	にぎやかな場が好き？ ←→ 一人の時間のほうが落ち着く？
Q6	S-N	今の現実を大事にする？ ←→ 未来や可能性を考える？
Q7	T-F	はっきり意見を伝える方？ ←→ 相手の気持ちを優先する方？
Q8	J-P	予定を立てるのが得意？ ←→ その場の流れで動きたい？
Q9	E-I	話すことで元気になる？ ←→ 話すと疲れる？
Q10	S-N	実体験や事実をもとに話す？ ←→ たとえ話や例が多い？
Q11	T-F	まず事実を整理してから判断する？ ←→ 感情の動きに従う？
Q12	J-P	物事をきっちり終わらせたい？ ←→ とりあえず動いてみたい？
Q13	E-I	大人数の中でも自分を出せる？ ←→ 少人数の方が安心？
Q14	S-N	現実的な人だと言われる？ ←→ 空想的だと言われる？

Q15	T-F	冷静だねと言われる？ ←→ 優しいねと言われる？
Q16	J-P	決断は早い方？ ←→ 保留にしがち？

※ プレイヤーは各質問に対して0～4のスライダーで回答する。

- 0 = 明確に左タイプに当てはまる
- 4 = 明確に右タイプに当てはまる
- 2 = 中間・どちらとも言えない

### 7.1.2 スコア判定ロジック

- 各軸には2つの質問（スライダー値0～4）が紐づく
- 合計値 0～8 によって左右どちらの傾向かを判定する

合計値	判定
0～4	左側 (E, S, T, J)
5～8	右側 (I, N, F, P)
ちょうど4	補助値でブレーク

#### 補助値によるブレーク（score = 4 の場合）

- E-I 軸 → 社交性（personality.social）  $\geq 4 \rightarrow E$ 、 $<4 \rightarrow I$
- S-N 軸 → 表現力（personality.expressiveness）  $\geq 4 \rightarrow N$ 、 $<4 \rightarrow S$
- T-F 軸 → 気配り（personality.kindness）  $\geq 4 \rightarrow F$ 、 $<4 \rightarrow T$
- J-P 軸 → 行動力（personality.activity）  $\geq 4 \rightarrow J$ 、 $<4 \rightarrow P$

### 7.1.3 判定スクリプト例

```
const determineType = (score, fallback) => {
  if (score < 4) return 'L';
  if (score > 4) return 'R';
  return fallback >= 4 ? 'L' : 'R';
};

const mbti = [
  determineType(q1 + q5, social) === 'L' ? 'E' : 'I',
  determineType(q2 + q6, expressiveness) === 'L' ? 'S' : 'N',
  determineType(q3 + q7, kindness) === 'L' ? 'T' : 'F',
  determineType(q4 + q8, activity) === 'L' ? 'J' : 'P'
].join('');
```

### 7.1.4 保存形式

- `mbti` : 4文字タイプ (例: "INFP")
- `mbti_slider` : 16項目 (Q1~Q16) のスライダー値 (0~4) を順に格納した配列
- 補助値 (社交性など) は `personality` フィールドに含まれており、外部参照可能

MBTIは以下の4軸 (8タイプ) から成り立つ:

軸名	左	右
E-I	外向 (Extraversion)	内向 (Introversion)
S-N	現実 (Sensing)	直感 (iNtuition)
T-F	思考 (Thinking)	感情 (Feeling)
J-P	判断 (Judging)	知覚 (Perceiving)

診断方式:

- 各軸に 2問ずつ、計16問のスライダー質問を用意
- スライダーは 5段階 (0~4) で、左寄りが前者、右寄りが後者に近いことを示す

#### スコア計算：

- 各軸ごとに2つのスライダー値（例：E-Iは質問1と5）を合計し、範囲0～8のスコアとする
- 0～4の場合は左側（E/S/T/J）、5～8は右側（I/N/F/P）と判定
- ちょうど4のときは性格値（行動力や社交性）によってブレイク：
  - 例：社交性4以上 → E、未満 → I

#### MBTI構築例：

```
const score_EI = q1 + q5;
const score_SN = q2 + q6;
const score_TF = q3 + q7;
const score_JP = q4 + q8;

MBTI = (score_EI >= 5 ? 'I' : 'E') +
      (score_SN >= 5 ? 'N' : 'S') +
      (score_TF >= 5 ? 'F' : 'T') +
      (score_JP >= 5 ? 'P' : 'J');
```

#### 保存形式：

- `mbti`：例「INFP」
- `mbti_slider`：質問ごとのスライダー値（0～4）×16個の配列

---

## 7.2 診断UIと手動設定

MBTIの性格タイプは、キャラクター追加時にプレイヤーが設定する。UI上では2種類の方式を切り替えて選択できる。

### 7.2.1 設定方式の選択

- 診断方式（推奨）
  - 16問のスライダー質問に回答して、MBTIを自動判定する
  - 判定に使用されたスライダー値（`mbti_slider`）は保存対象となる

- **手動選択方式**

- MBTIタイプ（4文字）をプルダウンから直接選択する
- 診断をスキップしたい場合や意図的に設定したいときに選択可能
- `mbti_slider` は空のままでも問題ない（診断未実施扱い）

### 7.2.2 UI仕様

- キャラ追加画面では、初期状態で「診断方式」が選択されている
- 上部タブまたはラジオボタンにて「診断方式／手動選択方式」の切り替えが可能
- 診断方式では、16問のスライダー質問フォームが出現
  - 各スライダーはラベル付き（例：「具体的 ←→ 抽象的」）
  - スライダー値の初期値は「2（中間）」に設定されている
- 手動方式では、以下のUI構成：
  - MBTI 4文字を個別に選択（E/I, S/N, T/F, J/P）
  - またはプリセットからまとめて選択（例：「INFP」など）

### 7.2.3 設定後の扱い

- 診断方式・手動方式のどちらで設定したかは保存上は問わない（`mbti` が主データ）
- 一度設定したMBTIは基本的に**キャラクター作成後は編集不可**
  - 編集が必要な場合は「キャラ編集」画面から手動で変更（管理者操作）
- `mbti_slider` の有無により、診断を行ったかどうかを判断できる（診断分析等に応用可）

### 7.2.4 補足

- 診断中にページを離れた場合、途中のスライダー値は保存されない

- 手動方式であっても、後から診断結果に基づいて再設定（再診断）は可能（任意）
  - `mbti_slider` が未入力かつ `mbti` のみが存在する場合は、手動設定されたものと見なす
  - ユーザーによるキャラ診断の手間を軽減するため、今後「性格プリセット選択（例：熱血タイプ）」の導入も検討中
  - キャラクター追加時、性格タイプは以下のいずれかで設定可能：
    - **診断方式**（スライダー回答）
    - **手動選択方式**（MBTI4文字を直接選択）
  - UI上で切替可能、どちらで設定したかは保存上の違いなし
  - `mbti_slider` は診断方式でのみ値が記録され、手動選択時は空のままでも可
- 

## 7.3 備考

- スライダー質問の文言・順序・分類は [7.1節](#) に記載されており、本仕様書内で完結する。
- MBTIに基づく性格タイプは、今後のイベント発生傾向・セリフスタイル補正などにも活用予定。
- 診断後のMBTIは、UI上では確認のみ可能であり、基本的に再編集は不可（管理操作による変更のみ可能）。