三目並べ詳細仕様書

目次

[1 概要 1](#_Toc168559434)

[2 用語の定義 1](#_Toc168559435)

[3 ファイル構成 1](#_Toc168559436)

[3.1 game\_manager.c 1](#_Toc168559437)

[3.2 display.c、display.h 1](#_Toc168559438)

[3.3 select.c、select.h 3](#_Toc168559439)

[4 全体の流れ 4](#_Toc168559440)

[5 機能構成・関数説明 5](#_Toc168559441)

[5.1 全体制御　main() 5](#_Toc168559442)

[5.2 表示機能 5](#_Toc168559443)

[5.3 選択機能 14](#_Toc168559444)

[6 画面構成 16](#_Toc168559445)

[6.1 開始・終了選択画面 17](#_Toc168559446)

[6.2 盤面 18](#_Toc168559447)

[6.3 結果画面 19](#_Toc168559448)

[6.4 エラー画面 20](#_Toc168559449)

[6.5 画面遷移 22](#_Toc168559450)

# 概要

本仕様書は用語の定義、ファイル構成、全体の流れ、機能構成、画面構成・関数説明を記述している。なお、本仕様書はc言語で記述すること想定している。また、エンコードの際、文字コードはUnicodeを用いることを想定している。

# 用語の定義

* 1. プレイヤー：操作し遊ぶ人。
  2. 記号：「〇」「×」を示す。

# ファイル構成

## game\_manager.c

* + - * display.hをインクルード
      * select.hをインクルード
      * 関数定義　main()：全体制御

## display.c、display.h

* + 1. display.c：
       - display.hをインクルード
       - 関数定義　GameBoard()：盤面表示
         * 関数定義　WhoTurn()：手番制御
         * 関数定義　InputBoard()：盤面入力制御
         * 関数定義　DisplayBoard()：盤面出力
         * 関数定義　JudgeMatch()：勝利判定
       - 関数定義　Result()：結果表示
       - 関数定義　InputError() : エラー表示
       - 関数定義　SameInputError() : 重複エラー表示
    2. display.h
       - 列挙型定義

|  |  |
| --- | --- |
| RESULT  試合結果 | WINNER\_A |
| WINNER\_B |
| DRAW…引き分け |
| NONE\_WINNER…試合続行 |
| R\_RESET |
|  |  |
| PLAYER\_TURN  手番 | TURN\_A |
| TURN\_B |
| P\_RESET |

* + - * マクロ定義

|  |  |
| --- | --- |
| RESET\_COMMAND | r |
| BOARD\_CELL | 3 |

* + - * プロトタイプ宣言　GameBoard()
      * プロトタイプ宣言　WhoTurn()
      * プロトタイプ宣言　InputBoard()
      * プロトタイプ宣言　DisplayBoard()
      * プロトタイプ宣言　JudgeMatch()
      * プロトタイプ宣言　Result()
      * プロトタイプ宣言　InputError()
      * プロトタイプ宣言　SameInputError()

## select.c、select.h

* + 1. select.c
       - select.hをインクルード
       - 関数定義　StartEnd()：開始・終了選択
       - 関数定義　Rematch()：再戦選択
    2. Select.h
       - 列挙型SELECTを定義

|  |  |
| --- | --- |
| SELECT  開始終了選択 | START |
| END |

* + - * プロトタイプ宣言　StartEnd()
      * プロトタイプ宣言　Rematch()

# 全体の流れ

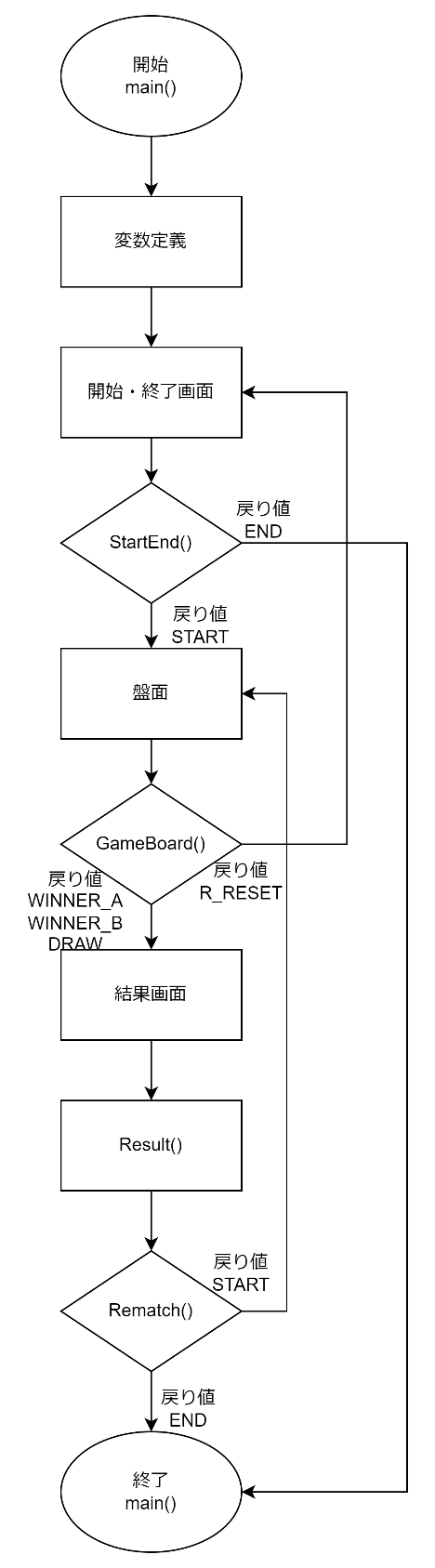


図1　全体フローチャート

# 機能構成・関数説明

本文内で明記されている変数以外に必要な変数があれば、各関数内で自由に定義し使用してもよい。マクロ定義する場合も同様である。ただしグローバル変数は使わないものとする。

## 全体制御　main()

* main()はgame\_manager.c内で記述する。
* GameBoard()、Result()、StartEnd()、Rematch()を用いて、ゲーム全体を制御する。

## 表示機能

* + 1. 盤面表示　GameBoard()
       - 型：RESULT
       - 引数：void
       - 戻り値：RESULT
       - 変数



* + - * 本関数はdisplay.cで定義し、main()内で呼び出す。
      * JudgeMatch()の戻り値がNONE\_WINNER以外になるまで、 WhoTurn()、DisplayBoard()、InputBoard()を用いて三目並べを進行する。
      * 試合結果を戻り値で返す。

ダイアグラム が含まれている画像

自動的に生成された説明

図2 GameBoard()のフローチャート

* + - * WhoTurn()
        + 型：PLAYER\_TURN
        + 引数：PLAYER\_TURN
        + 戻り値：PLAYER\_TURN
        + 本関数はdisplay.cで定義し、Gameboard()内で呼び出す。
        + 初めてのループか判定し、初めてならばランダムで手番決定。
        + 前回の手番の人を確認し、手番を交代する。
        + 手番を表示する。表示レイアウト6.2盤面を参照。
        + 遊ぶ人は「プレイヤーA」「プレイヤーB」の二人。
        + 決定した手番を戻り値で返す。
        + 初めてのループならばP\_RESET、Aが手番のときTURN\_A、Bが手番のときTURN\_Bとする。

名刺, テキスト が含まれている画像

自動的に生成された説明

* + - * + 図3 WhoTurn()のフローチャート
      * DisplayBoard()
        + 型：void
        + 引数：配列boardのアドレス
        + 戻り値：void
        + 本関数はdisplay.cで定義し、Gameboard()内で呼び出す。
        + 配列の中身を表示する。表示レイアウトは6.2盤面を参照。

テキスト

自動的に生成された説明

図4 DisplayBoardのフローチャート

* + - * InputBoard()
        + 型：PLAYER\_TURN
        + 引数：PLAYER\_TURN、配列boardのアドレス
        + 戻り値：PLAYER\_TURN
        + 本関数はdisplay.cで定義し、Gameboard()内で呼び出す。
        + プレイヤーに入力を求めるテキストを表示。ゲームをリセットするためのコマンドを表示。表示レイアウトは6.2盤面を参照。
        + プレイヤーの入力を取得。半角一文字まで取得。一文字を超える場合はInputError()でエラーメッセージを表示し、再入力をユーザに求める。
        + 半角数字1～9とRESET\_COMMAND以外の入力はエラーとし、InputError ()でエラーメッセージを表示し、再入力をユーザに求める。
        + 入力なしの場合InputError ()でエラーメッセージを表示し、再入力をユーザに求める。
        + 入力された文字が1～9の文字ならば、配列の中から同じ文字を探し出し、その文字が格納されている配列のアドレスを取得する
        + 取得したアドレスから配列の中を参照し、記号で上書きする
        + 既に記号が書き込まれている場所の数字を入力した場合SameInputError()で重複入力エラーメッセージを出力し、再度入力を求める。

タイムライン が含まれている画像

自動的に生成された説明

図5 InputBoard()のフローチャート

* + - * + 記号「〇」「×」はchar型の半角文字「o」「x」を使用する。
        + プレイヤーAが「〇」、プレイヤーBが「×」とする。
        + 入力された文字がRESET\_COMMAND文字ならば戻り値P\_RESETを返し、処理を抜ける。
        + 誰の手番で記号上書きが終了したかを戻り値で返す。
      * JudgeMatch()
        + 型：RESULT
        + 引数：PLAYER\_TURN、配列boardのアドレス
        + 戻り値：RESULT
        + 本関数はdisplay.cで定義し、Gameboard()内で呼び出す。
        + 記号の探索は引数で受け取ったプレイヤーの記号を探索する
        + 勝利判定は、同じ記号が縦、横、斜めに３つ並んでいるのが一つでも存在するとき、引数で受け取ったプレイヤーの勝利とし、プレイヤーAの勝利ならWINNER\_A、プレイヤーBの勝利ならWINNER\_Bを戻り値として返す
        + 引き分けは、盤面すべてが記号で埋まり３つ同じ記号が一つもそろっていない場合にDRAWを戻り値として返す
        + 盤面すべてが記号で埋まっておらず、3つ同じ記号が一つもそろっていない場合にNONE\_WINNERを返す
        + 判定結果を戻り値で返す。

ダイアグラム

自動的に生成された説明

図6 JudgeMatch()のフローチャート

* + 1. 結果表示　Result()
       - 型：void
       - 引数：RESULT
       - 戻り値：void
       - 変数



* + - * 本関数はdisplay.cで定義し、main()内で呼び出す。
      * 引数で試合の状態を受け取り、WINNER\_Aなら「Aの勝利」、WINNER\_ Bなら「Bの勝利」、DRAWなら「引き分け」とテキストで表示する。R\_RESETなら、何も表示しない
      * WINNER\_Aを受け取ると、Aの勝利数をインクリメントし、「Aの勝利回数：勝利数」を表示する。WINNER\_Bの場合は、Bの勝利数をインクリメントし「Bの勝利回数：勝利数」を表示する。

ダイアグラム

自動的に生成された説明

図7 Resultのフローチャート

* + 1. エラー表示　InputError()
       - 型：void
       - 引数：void
       - 戻り値：void
* 本関数はdisplay.cで定義し、エラー処理が必要な関数で呼び出す。
* 不正な入力があったときにエラー画面を表示する関数。開始・終了選択画面、盤面、結果画面で使用する。表示画面の構成は6.4.1を参照。

テキスト が含まれている画像

自動的に生成された説明

図8 InputError()のフローチャート

* + 1. 重複エラー表示　SameInputError()
       - 型：void
       - 引数：void
       - 戻り値：void
* 本関数はdisplay.cで定義し、InputBoard()で呼び出す。
* 1~9の入力が重複したときにエラー画面を表示する関数。盤面で使用する。表示画面の構成は6.4.2を参照。

テキスト が含まれている画像

自動的に生成された説明

図9 SameInputError()のフローチャート

## 選択機能

* + 1. 開始・終了選択　StartEnd()
       - 型：SELECT
       - 引数：void
       - 戻り値：SELECT
       - 本関数はselect.cで定義し、main()内で呼び出す。
       - ユーザにゲームの開始終了を[y/n]で選択させる。
       - yで三目並べ開始、nで三目並べを終了する。
       - [y/n]以外の入力はInputError ()でエラーメッセージを表示し、再入力をユーザに求める。
       - 戻り値はyのときSTART、nのときENDを返す。

ダイアグラム

中程度の精度で自動的に生成された説明

図10 StartEnd()のフローチャート

* + 1. 再戦選択　Rematch()
       - 型：SELECT
       - 引数：void
       - 戻り値：SELECT
       - 本関数はselect.cで定義し、main()内で呼び出す。
       - ユーザにゲームの再戦を[y/n]で選択させる。
       - yで三目並べ再戦、nで三目並べを終了する。
       - [y/n]以外の入力はInputError ()でエラーメッセージを表示し、再入力をユーザに求める。
       - 戻り値はyのときSTART、nのときENDを返す。

ダイアグラム

中程度の精度で自動的に生成された説明

図11 Rematch()のフローチャート

# 画面構成

・画面内の平仮名と漢字は全角、アルファベットと数字と記号は半角で表記すること

・事前にプレイヤーの入力を求める場所から画面上部を表示しておき、入力後に画面下部を表示させる

## 開始・終了選択画面

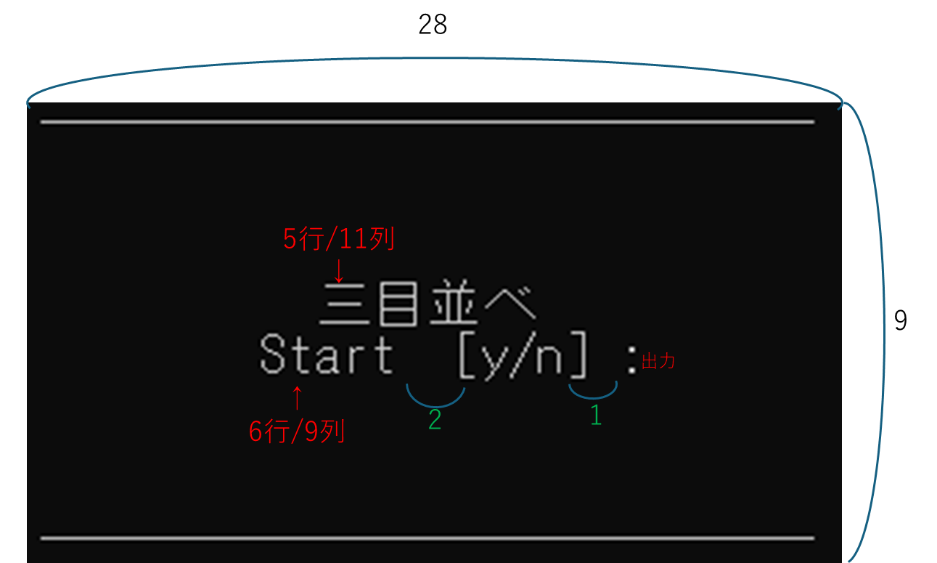


図12　開始・終了選択画面

・1行目と９行目に「\_」を横28列分記載

・5行目の11列目に「三目並べ」と記載

・6行目の9列目に「Start」と記載

・「Start」からスペースを2つ分空け、「[y/n]」と記載

・「[y/n]」からスペースを1つ分空け、「:」と記載

・「:」の右にプレイヤーが入力した文字が出力されるようにする

・yの入力で盤面へ行く、nの入力で終了する

## 盤面

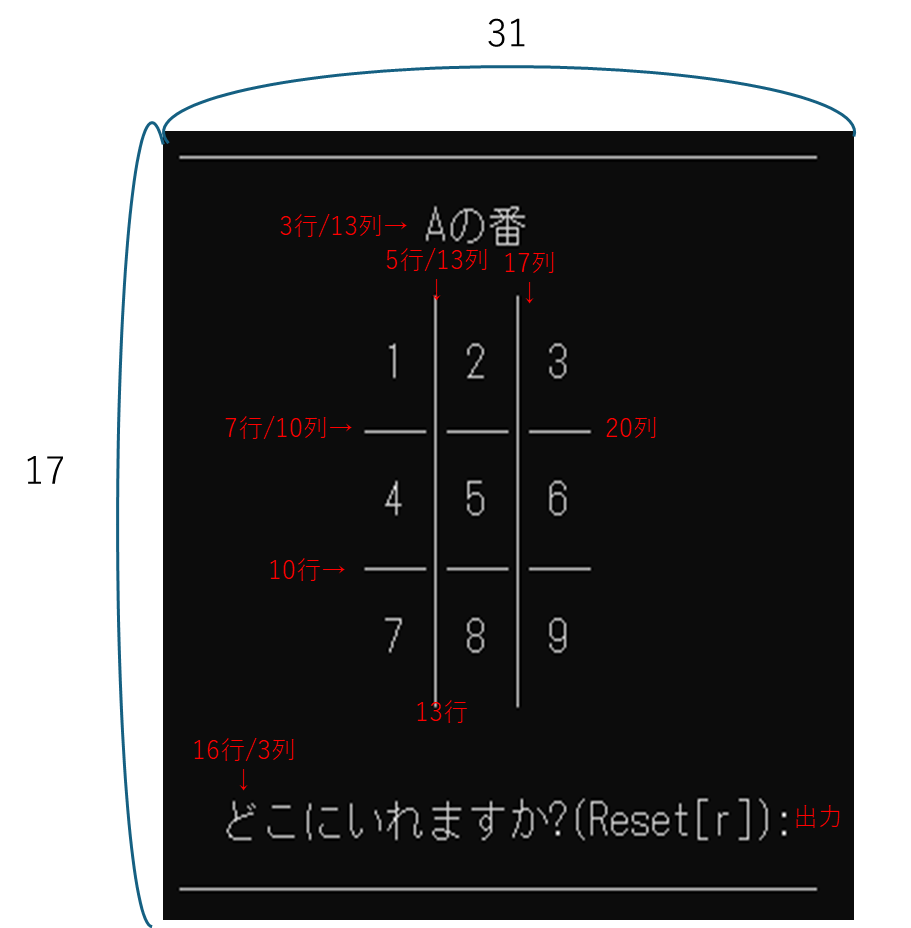


図13　盤面

・1行目と16行目に「\_」を横24列分記載

・3行目の13列目に手番に応じ、「Aの番」、「Bの番」と記載

・5行目の13列目と17列目から縦に13行目まで「|」を記載し、盤面の縦線を描く

・7行目と10行目の10列目から横に20列目まで「\_」を記載し、盤面の横線を描く(「|」は置き換えない)

・ 盤面の縦線と横線によって内側が枠に囲まれている。外側も縦線と横線から同じ枠があると仮定し、これらの中心にそれぞれ1~9の整数を記載する。整数は左上の1から右に1つずつ大きくなり、3の倍数ごとに枠の単位で改行し、最後の右下は9となる

・盤面の整数は変数で定義し、記号の変化にも対応できるようにする

・16行目の3列目に「どこにいれますか?(Reset[r]):」を記載

・「：」の右にプレイヤーが入力した文字が出力されるようにする

・入力した数値と同じ盤面の数値を手番に応じて「○」「×」を代入

・「R」を入力した際、開始・終了画面へ戻る

## 結果画面

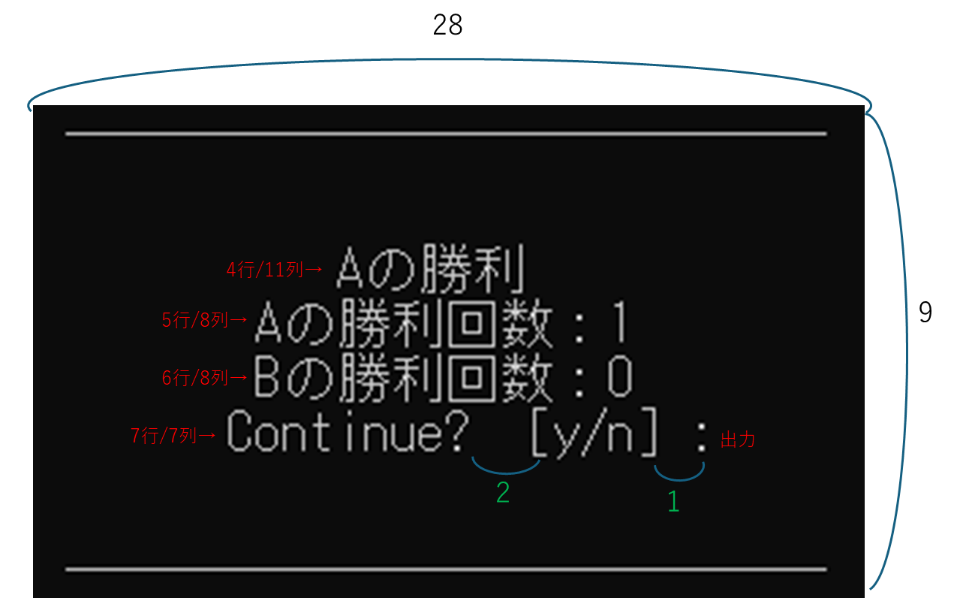


図14　結果画面

・1行目と9行目に「\_」を横28列分記載

・4行目の11列目に結果に応じて「Aの勝利」、「Bの勝利」、「引き分け」と記載

・5行目の8列目に「Aの勝利回数：」と記載

・6行目の8列目に「Bの勝利回数：」と記載

・ 「Aの勝利回数：」、 「Bの勝利回数：」の右に勝利回数だけ＋１されるように表記(初期値は0)

・ 7行目の7列目に「Continue?」と記載

・「Continue?」からスペースを2つ分空け、「[y/n]」と記載

・「[y/n]」からスペースを1つ分空け、「:」と記載

・「：」の右にプレイヤーが入力した文字が出力されるようにする

・ yの入力で盤面へ行く、nの入力で終了する

## エラー画面

* + 1. 不正な入力

テキスト

自動的に生成された説明

　　　　　　　　　　　　図15　不正な入力のエラー画面

・両端を空けて「\_」を横24列、これを1行目と5行目に記載

・「|」を2~5行目の左右両端に記載

・3行目の7列目に「不正な入力です」と記載

・4行目の2列目に「もう一度入力してください」と記載

・開始・終了画面や結果画面のエラー、盤面で1~9、R以外を入力した時のエラーに用いる。

・表示後にエラー前と同じ再入力を求める。

　　　5.4.2　1~9重複のエラー

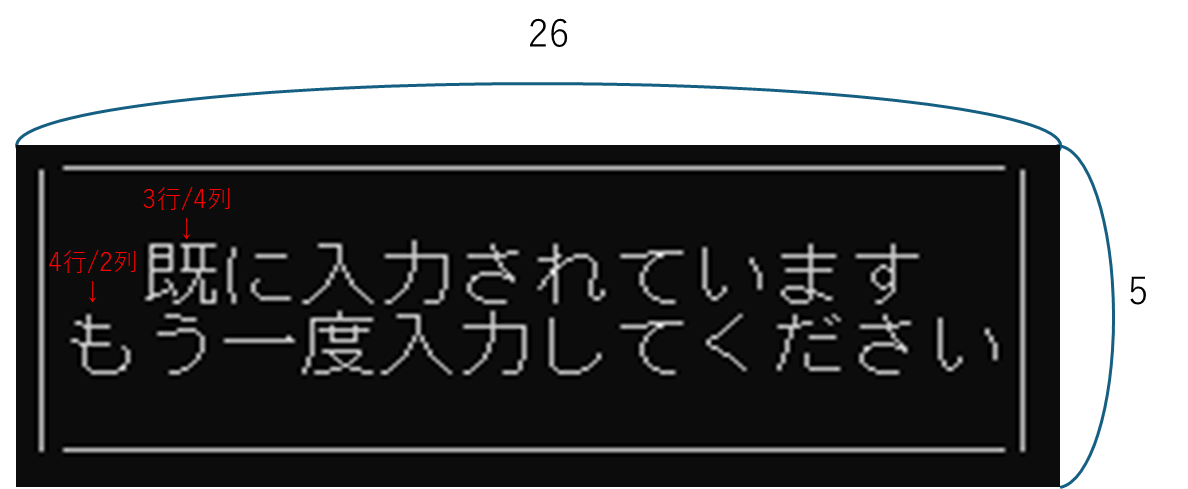


　　　　　　　　　　　　図16　1~9重複のエラー画面

・両端を空けて「\_」を横24列、これを1行目と5行目に記載

・「|」を2~5行目の左右両端に記載

・3行目の4列目に「既に入力されています」と記載

・4行目の2列目に「もう一度入力してください」と記載

・盤面で1~9の入力が重複したときのエラーに用いる

・表示後にエラー前と同じ再入力を求める。

## 画面遷移

