三目並べ詳細仕様書

目次

[1 概要 1](#_Toc167270834)

[2 言葉の定義 1](#_Toc167270835)

[3 ファイル構成 1](#_Toc167270836)

[3.1 game\_manager.c 1](#_Toc167270837)

[3.2 display.c、display.h 1](#_Toc167270838)

[3.3 select.c、select.h 2](#_Toc167270839)

[3.4 全体の流れ 3](#_Toc167270840)

[4 機能構成 4](#_Toc167270841)

[4.1 表示機能 4](#_Toc167270842)

[4.2 選択機能 6](#_Toc167270843)

[5 画面構成 7](#_Toc167270844)

[5.1 開始・終了選択画面 7](#_Toc167270845)

[5.2 盤面 8](#_Toc167270846)

[5.3 結果画面 9](#_Toc167270847)

[5.4 エラー画面 10](#_Toc167270848)

[5.5 画面遷移 12](#_Toc167270849)

# 概要

本仕様書の言葉の定義、ファイル構成、機能構成、画面構成の説明。

# 言葉の定義

* 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

* + - * main()：全体制御

## display.c、display.h

* + 1. display.c
       - GameBoard()：盤面表示
         * WhoTurn()：手番制御
         * InputBoard()：盤面入力制御
         * DisplayBoard()：盤面出力
         * JudgeMatch()：勝利判定
       - Result()：結果表示
       - InputError() : エラー表示
       - SameInputError() : 重複エラー表示
    2. display.h
       - GameBoard()
       - WhoTurn()
       - InputBoard()
       - DisplayBoard()
       - JudgeMatch()
       - Result()
       - InputError()
       - SameInputError()

## select.c、select.h

* + 1. select.c
       - StartEnd()：開始・終了選択
       - Rematch()：再戦選択
    2. Select.h
       - StartEnd()
       - Rematch()

## 全体の流れ

ダイアグラム

自動的に生成された説明

図1　全体フローチャート

# 機能構成

* 列挙型の定義

|  |  |
| --- | --- |
| RESULT  試合結果 | WINNER\_A |
| WINNER\_B |
| DRAW |
| NONE\_WINNER |
| R\_RESET |
|  |  |
| PLAYER\_TURN  手番 | TURN\_A |
| TURN\_B |
| P\_RESET |
|  |  |
| SELECT  開始終了選択 | STRAT |
| END |

引き分け

試合続行

* マクロの定義

|  |  |
| --- | --- |
| RESET\_COMMAND | r |

## 表示機能

* + 1. 盤面表示　GameBoard()
       - 型：RESULT
       - 引数：void
       - 戻り値：RESULT
       - 勝者、引き分け、リセットが出るまで以下の関数を用いて、三目並べを進行
       - WhoTurn()
         * 型：PLAYER\_TURN
         * 引数：PLAYER\_TURN
         * 戻り値：PLAYER\_TURN
         * 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
         * 遊ぶ人は「プレイヤーA」「プレイヤーB」の二人
       - DisplayBoard()
         * 型：void
         * 引数：配列boardのアドレス
         * 戻り値：void
         * 3\*3配列に1～9の文字をboard [0][0]から数字が小さい順に格納
         * 配列の中身を、画面に表示する
       - InputBoard()
         * 型：PLAYER\_TURN
         * 引数：PLAYER\_TURN、配列boardのアドレス
         * 戻り値：PLAYER\_TURN
         * 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
         * プレイヤーの入力を取得
         * 半角数字1～9とRESET\_COMMAND以外の入力はエラーとし、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
         * 記号「〇」「×」はchar型の半角文字「o」「x」を使用する
         * 入力された文字がRESET\_COMMAND文字ならば配列の初期化処理をして、戻り値RESETを返し、処理を抜ける
         * 初期化処理は1～9の文字を[0][0]から数字が小さい順に格納する
       - JudgeMatch()
         * 型：RESULT
         * 引数：PLAYER\_TURN、配列boardのアドレス
         * 戻り値：RESULT
         * 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する
         * 引き分けは、盤面すべてが埋まり３つ同じ記号が一つもそろっていないとき
    2. 結果表示　Result()
       - 型：void
       - 引数：RESULT
       - 戻り値：void
       - 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。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、A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。
       - AとBの勝利回数の変数をそれぞれcount\_a\_win, count\_b\_winとする。
    3. エラー表示　InputError()
       - 型：void
       - 引数：void
       - 戻り値：void
* 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・終了画面、盤面、結果画面で使用する。表示画面の構成は5.4.1を参照。
  + 1. 重複エラー表示　SameInputError()
       - 型：void
       - 引数：void
       - 戻り値：void
* 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5.4.2を参照。

## 選択機能

* + 1. 開始・終了選択　StartEnd()
       - 型：SELECT
       - 引数：void
       - 戻り値：SELECT
       - 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]以外の入力はエラーとし、再入力をユーザに求める。
       - 戻り値はユーザの選択の結果を返す。
    2. 再戦選択　Rematch()
       - 型：SELECT
       - 引数：void
       - 戻り値：SELECT
       - 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]以外の入力はエラーとし、再入力をユーザに求める。

# 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

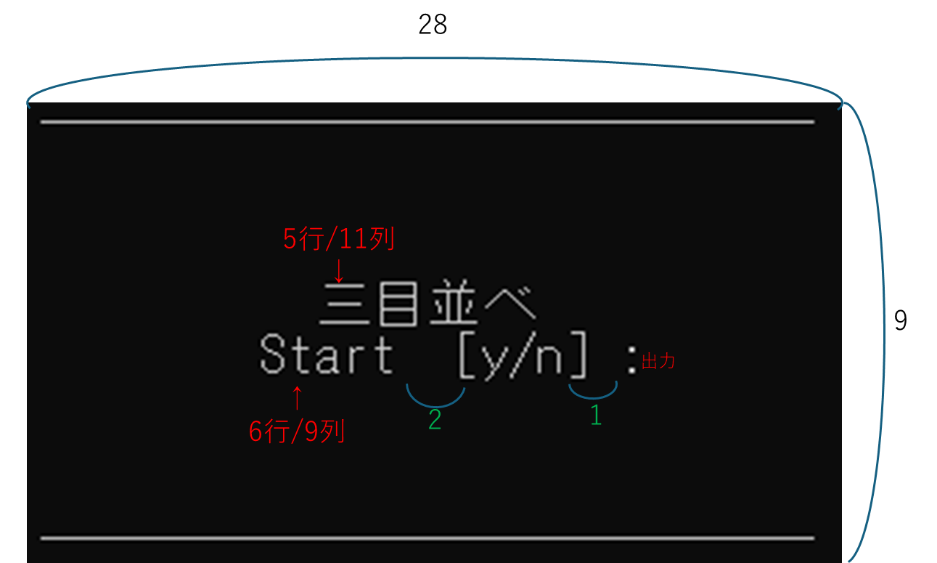


図2　開始・終了選択画面

・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

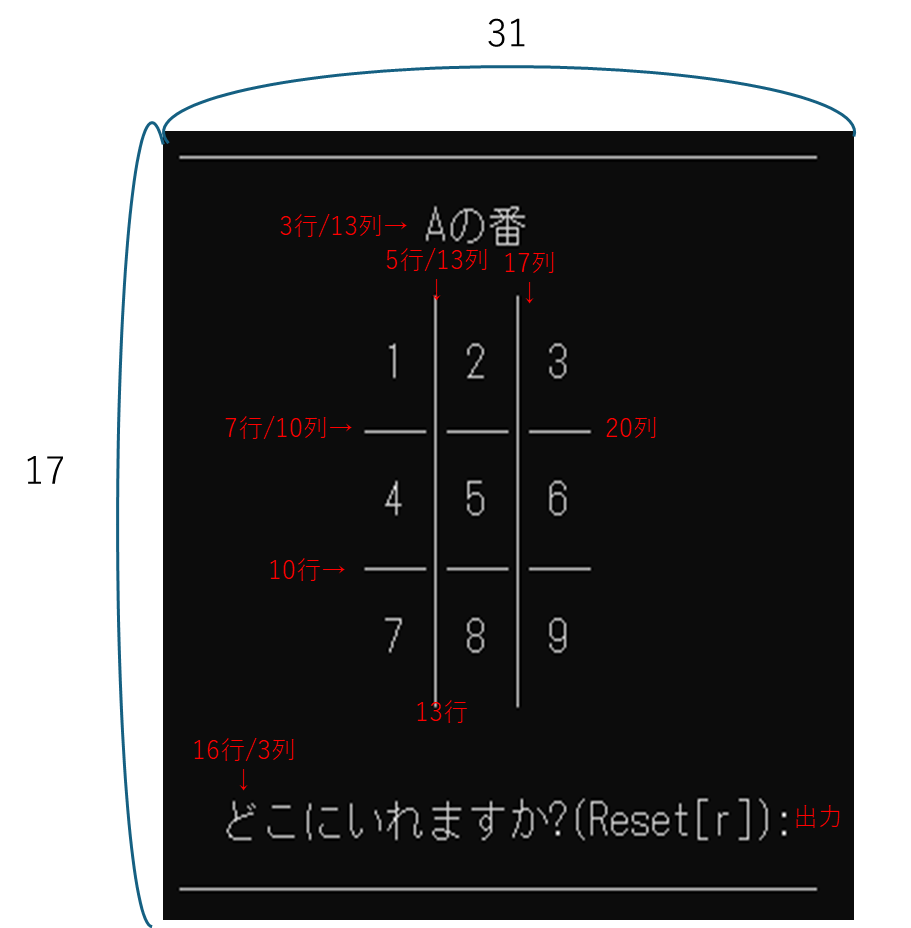


図3　盤面

・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

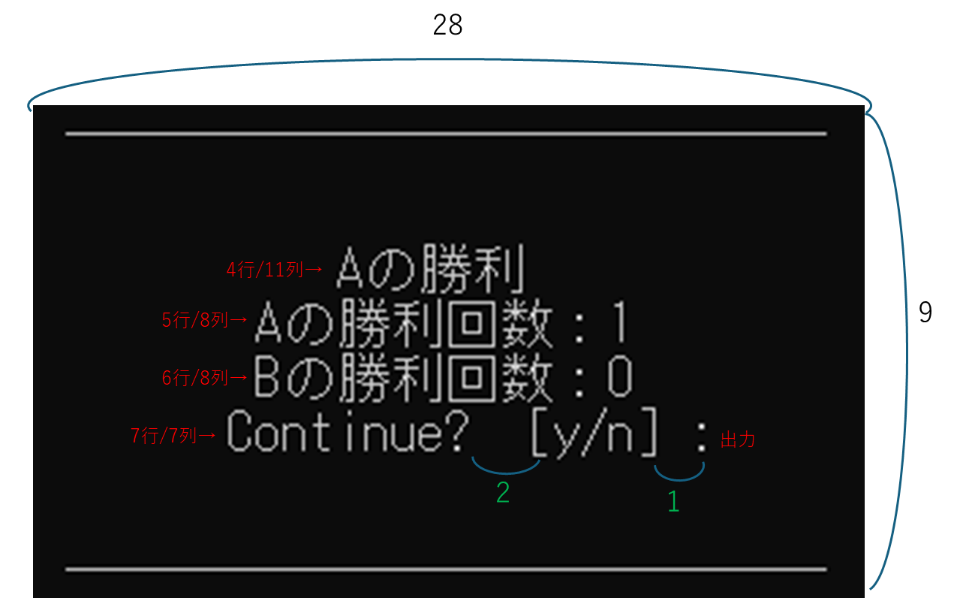


図4　結果画面

・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

・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

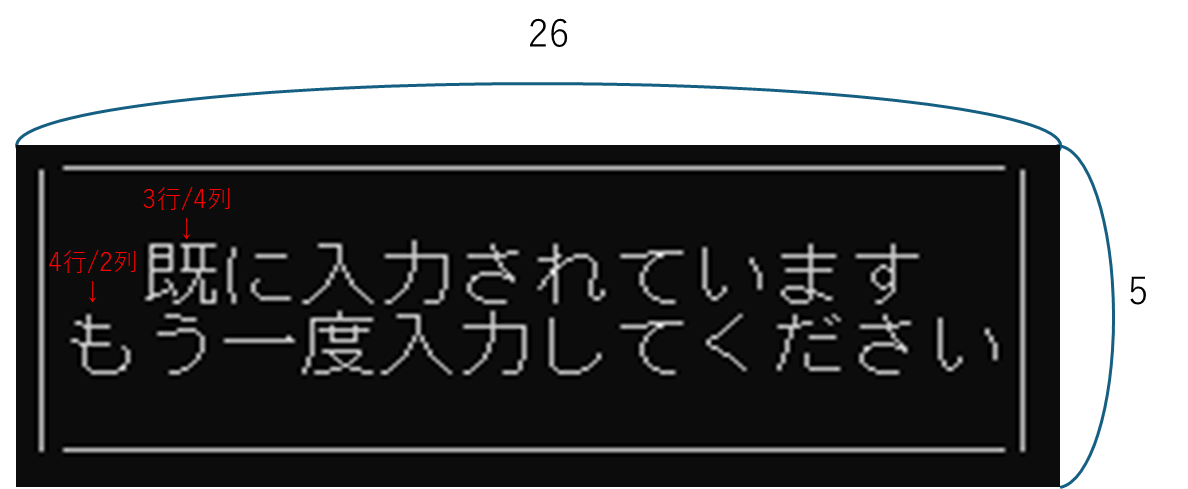
・3行目の7列目に「不正な入力です」と記載

・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



・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

