三目並べ詳細仕様書

Iチームメンバー

杉友泰宣　藤井健

目次

[1. 概要 5](#_Toc168496607)

[2. 画面構成 5](#_Toc168496608)

[2.1 画面詳細 5](#_Toc168496609)

[2.2 盤面表示画面 6](#_Toc168496610)

[2.2.1. 実行すると「o」、先攻、「x」、後攻、三目並べ画面、手番、「縦の値を入力してください」が画面に表示される。 6](#_Toc168496611)

[2.2.2. 縦の値を入力した後に「横の値を入力してください：」を表示する。 6](#_Toc168496612)

[2.2.3. 「o」「x」が出力された三目並べ画面が出力される 6](#_Toc168496613)

[2.3 結果表示画面 7](#_Toc168496614)

[勝利画面表示 7](#_Toc168496615)

[引き分け画面表示 7](#_Toc168496616)

[2.4 終了選択画面 8](#_Toc168496617)

[もう一度遊ぶ…0を選んだ場合 8](#_Toc168496618)

[2.5 エラー出力画面 8](#_Toc168496619)

[2.5.1 縦の値、横の値が1～3の半角整数値でない場合のエラー文 8](#_Toc168496620)

[2.5.2 既に入力されている箇所を指定した場合のエラー文 8](#_Toc168496621)

[3. ファイル構成 9](#_Toc168496622)

[3.1 tic\_tac\_toe.c 10](#_Toc168496623)

[3.2 get\_input.c 10](#_Toc168496624)

[3.3 get\_output.c 10](#_Toc168496625)

[3.4 get\_judge.c 11](#_Toc168496626)

[4. 機能構成 12](#_Toc168496627)

[main()関数 12](#_Toc168496628)

[Mainで使う変数 14](#_Toc168496629)

[Input\_Reset()関数 14](#_Toc168496630)

[Input\_Number()関数 15](#_Toc168496631)

[Input\_Mark()関数 15](#_Toc168496632)

[Output\_Screen()関数 16](#_Toc168496633)

[Output\_Result()関数 16](#_Toc168496634)

[Judge\_Turn()関数 17](#_Toc168496635)

[Judge\_Range()関数 17](#_Toc168496636)

[Judge\_Mark()関数 17](#_Toc168496637)

[Judge\_Result()関数 18](#_Toc168496638)

[Judge\_Retry()関数 19](#_Toc168496639)

[ターンを表す列挙型 19](#_Toc168496640)

[真か偽を表す列挙型 19](#_Toc168496641)

[三目並べの続行を表す列挙型 19](#_Toc168496642)

[勝利を表す列挙型 20](#_Toc168496643)

[マスが埋まっているかを表すマクロ定義 20](#_Toc168496644)

[入力文字列の要素数を表すマクロ定義 20](#_Toc168496645)

[列数を表すマクロ定義 20](#_Toc168496646)

[行数を表すマクロ定義 20](#_Toc168496647)

# 概要

本詳細仕様書は、画面構成、ファイル構成、機能構成を図や説明を記載する。

# 画面構成

### 画面詳細

数字と記号は全て半角で表示する。

記号は「o」（オー）と「x」（エックス）で表示する。

マスは縦区切りの’|’（バーティカルバー）でマスの表示をする。

ダイアグラム, 図形

自動的に生成された説明

マスの上に半角スペース４つの後ろに「o」を表示し、「o」の後ろに半角スペースを２つ入れ、先攻と表示する。またその次の行に「o」と先攻と同じように「x」と後攻を表示する。盤面は１マス半角スペース二つで表示し、数字または記号がある場合は前に半角スペース、後ろに半角数字または記号を表示する。

ダイアグラム

自動的に生成された説明

### 盤面表示画面

##### 実行すると「o」、先攻、「x」、後攻、三目並べ画面、手番、「縦の値を入力してください」が画面に表示される。

テキスト

中程度の精度で自動的に生成された説明

##### 縦の値を入力した後に「横の値を入力してください：」を表示する。

電子機器, 回路, 時計 が含まれている画像

自動的に生成された説明

##### 「o」「x」が出力された三目並べ画面が出力される

電子機器, 回路, コンピュータ が含まれている画像

自動的に生成された説明

### 結果表示画面

勝利画面表示

図形 が含まれている画像

自動的に生成された説明

引き分け画面表示

図形 が含まれている画像

自動的に生成された説明

### 終了選択画面

もう一度遊ぶ…0を選んだ場合

矢印 が含まれている画像

自動的に生成された説明

### エラー出力画面

1. 縦の値、横の値が1～3の半角整数値でない場合のエラー文

グラフィカル ユーザー インターフェイス が含まれている画像

自動的に生成された説明

1. 既に入力されている箇所を指定した場合のエラー文

グラフィカル ユーザー インターフェイス が含まれている画像

自動的に生成された説明

# ファイル構成

文字の書かれた紙

自動的に生成された説明

## tic\_tac\_toe.c

|  |  |  |
| --- | --- | --- |
| 型 | 関数名 | 説明 |
| Int | main | メインを記述 |

## get\_input.c

|  |  |  |
| --- | --- | --- |
| 型 | 関数名 | 説明 |
| void | Input\_Reset() | 盤面を初期化 |
| void | Input\_Number() | 座標の入力 |
| void | Input\_Mark() | 「o」「x」のどちらを入力 |

## get\_output.c

|  |  |  |
| --- | --- | --- |
| 型 | 関数名 | 説明 |
| void | Output\_Screen() | 三目並べ画面の出力 |
| void | Output\_Result() | 結果の出力 |

## get\_judge.c

|  |  |  |
| --- | --- | --- |
| 型 | 関数名 | 説明 |
| TURN(列挙型) | Judge\_turn() | 手番の判定 |
| BOOL(列挙型) | Judge\_Range() | 数値の範囲を判定 |
| BOOL(列挙型) | Judge\_Mark() | 「o」「x」の重複判定 |
| JUDGE(列挙型) | Judge\_Result() | 勝敗の判定 |
| EXIT(列挙型) | Judge\_Retry() | 続行判定 |

# 機能構成

1. tic\_tac\_toe.c

###### main()関数

ダイアグラム

自動的に生成された説明

三目並べの終了が出力されるまで回すループ

・Input\_Reset()関数、Judge\_Turn()関数、Output\_Screen関数の順で関数を呼び出し、処理を行う。

・Output\_Result()関数を用いて勝敗を表示し、Judge\_Retry()関数を用いてENDの時はループを終了し、CONTINUEの時は、再度ループを始める。

勝敗の結果が出力されるまで回すループ

・Output\_Screen()関数で、三目並べ画面を出力する。

・Judge\_Result()関数の戻り値がUNKNOWNの時、ターンを加算し再度ループを行う。O\_WIN、またはX\_WIN、DROWのときループが終了する。

入力の処理が終わるまでループ

・入力の処理としてInput\_Number ()関数、Judge\_Range()関数、Judge\_Mark()関数の順で関数を呼び出す。

・Judge\_Range()関数、Judge\_Mark()関数どちらもFALSEの場合、再度ループの処理を行う。TRUEの時にループから抜け出す。

###### Mainで使う変数

|  |  |  |
| --- | --- | --- |
| 変数 | 型 | 初期化 |
| ターンをカウントする変数 | int | 0 |
| 縦横の値を入力する一次元配列 [INDEX] | char | 0 |
| 盤面の情報を持つ二次元配列 [GRIT\_WIDE][GRIT\_HEIGHT] | char | 0 |
| 数値の範囲を判定した結果を 格納する変数 | BOOL | FALSE |
| 要素内の重複判定の結果を 格納する変数 | BOOL | FALSE |
| ターンの判定をした結果を格納する変数 | TURN | CIRCLE |
| 続行の判定をした結果を 格納する変数 | EXIT | CONTINUE |
| 勝敗の結果を格納する変数 | JUDGE | UNKNONWN |

1. get\_input.c

###### Input\_Reset()関数

|  |  |
| --- | --- |
| 引数 | (char型)盤面の情報を持つ２次元配列のポインタ[列数を表すマクロ] |
| 戻り値 | なし(void型) |
| 説明 | 配列の要素をすべて空白で代入する。 |

###### Input\_Number()関数

|  |  |
| --- | --- |
| 引数 | マスの座標を持つ一次元配列のポインタ(char型)、ターン数の情報を持つ変数(int型) |
| 戻り値 | なし(void型) |
| 説明 | メインでカウントしているターン数から手番を判別し、出力する。  1から3の整数値を配列の要素番号「0」に縦の数値を入力し、「1」に横の数値を入力する。 |
| 表示画面 | [2.2](#_盤面表示画面) |

###### Input\_Mark()関数

|  |  |
| --- | --- |
| 引数 | char\*型 盤面の情報を持つ二次元配列の要素[行数を表すマクロ][列数を表すマクロ]  int型 ターンの情報を持つ変数 |
| 戻り値 | CIRCLEまたはCROSSを格納する変数（） |
| 説明 | ターンの情報から「o」「x」のどちらの入力を行うのか判定する。 座標の情報 |
| 表示画面 | [2.2](#_表示画面) |

1. get\_output.c

###### Output\_Screen()関数

|  |  |
| --- | --- |
| 引数 | 盤面の情報を持つ２次元配列の要素[列数を表すマクロ](char型) |
| 戻り値 | なし(void型) |
| 説明 | 盤面の譲歩を持つ二次元配列の要素[行数][列数]と3×3のマス、先攻後攻と「o」「x」を出力する |
| 表示画面 | [2.1](#_画面詳細) |

###### Output\_Result()関数

|  |  |  |
| --- | --- | --- |
| 引数 | 勝敗の情報を持つ変数(int型) | |
| 戻り値 | なし(void型) | |
| 説明 | 勝敗の情報を持つ変数が勝利を表す列挙型のO\_WINまたはX＿WINまたはDROWのときそれぞれの結果を出力する。 | |
| 表示画面 | [勝利画面](#_勝利画面表示_1) | [引き分け画面](#_引き分け画面表示) |

1. get\_judge.c

###### Judge\_Turn()関数

|  |  |
| --- | --- |
| 引数 | ターン数の情報を持つ変数(int型) |
| 戻り値 | CIRCLEかCROSSを格納した変数 |
| 説明 | ターン数から手番を判定し偶数回ならCIRCLE、 奇数回ならCROSSを返す。 |

###### Judge\_Range()関数

|  |  |
| --- | --- |
| 引数 | マスの座標の情報を持つ一次元配列[2](char型) |
| 戻り値 | 真か偽を表す列挙型のTRUEまたはFALSEを代入した変数 |
| 説明 | 配列の数値が半角整数値1～3の範囲内かどうかを判別し、範囲内 であれば、TRUE、範囲外であれば警告文を出力し、FALSEを返す |
| 表示画面 | 警告文の表示…[2.5.1](#_縦の値、横の値が1～3の半角整数値でない場合のエラー文) |

###### Judge\_Mark()関数

|  |  |
| --- | --- |
| 引数 | マスの座標の情報を持つ一次元配列[2](char型)  char型 盤面の情報を持つ二次元配列[行数を表すマクロ][列数を表すマクロ] |
| 戻り値 | 真か偽を表す列挙型のTRUEまたはFALSEを代入した変数 |
| 説明 | マスの座標における配列の要素が空白であるときTRUEを返し、また空白でないときは警告文を出力し、FALSEを返す。 |
| 表示画面 | 警告文の表示…[2.5.2](#_既に入力されている箇所を指定した場合のエラー文) |

###### Judge\_Result()関数

|  |  |
| --- | --- |
| 引数 | 盤面の情報を持つ二次元配列[行数を表すマクロ][列数を表すマクロ](char型)  メインでカウントしているターン数の情報を持つ変数(int型) |
| 戻り値 | 勝利を表す列挙型のO\_WIN、X\_WIN、DRAW、UNKNOWNのいずれかを格納した変数 |
| 説明 | 「o」が縦、横、斜めのどれかに三つ並んだか判定し並んでいるときはO\_WIN、「x」のときはX\_WIN、ターン数がCAPACITY\_FULLの時DROW度の条件も満たさない場合はUNKOWNを 格納した変数を返す。 |

###### Judge\_Retry()関数

|  |  |
| --- | --- |
| 引数 | なし |
| 戻り値 | 三目並べ続行を表す列挙型のENDまたはCONTINUEを格納した変数 |
| 説明 | 「もう一度遊ぶ…０/やめる…０以外」で0以外を入力するとEND、０を入力するとCONTINUEを代入した変数を返す。 |
| 表示画面 | [もう一度遊ぶ](#_もう一度遊ぶ…0を選んだ場合_1) | |

1. define.h

ターンを表す列挙型

|  |  |  |
| --- | --- | --- |
| 型名 | TURN | |
| 列挙子 | CIRCLE:「o」の手番 | CROSS:「x」の手番 |

真か偽を表す列挙型

|  |  |  |
| --- | --- | --- |
| 型名 | BOOL | |
| 列挙子 | TRUE：真 | FALSE：偽 |

三目並べの続行を表す列挙型

|  |  |  |
| --- | --- | --- |
| 型名 | EXIT | |
| 列挙子 | CONTINUE：三目並べをもう一度遊ぶ | END：三目並べを終了する |

勝利を表す列挙型

|  |  |  |
| --- | --- | --- |
| 型名 | WIN | |
| 列挙子 | UNKNOWN：勝敗の結果が出ていない。 | DROW：引き分け |
| O\_WIN：「o」の勝利 | X\_WIN：「x」の勝利 |

マスが埋まっているかを表すマクロ定義

|  |  |
| --- | --- |
| 型名 | #define CAPACITY\_FULL 9 |
| 説明 | マスがすべて埋まっている。 | |

入力文字列の要素数を表すマクロ定義

|  |  |
| --- | --- |
| 型名 | #define INDEX 2 |
| 説明 | 入力文字列の要素数を表すマクロ定義。 | |

列数を表すマクロ定義

|  |  |
| --- | --- |
| 型名 | #define GRIT\_HEIGHT 9 |
| 説明 | マスがすべて埋まっている。 | |

行数を表すマクロ定義

|  |  |
| --- | --- |
| 型名 | #define GRIT\_WIDE 3 |
| 説明 | マスがすべて埋まっている。 | |