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](#_Toc168652640)

[2. 画面構成 5](#_Toc168652641)

[2.1 画面詳細 5](#_Toc168652642)

[2.2 盤面表示画面 6](#_Toc168652643)

[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](#_Toc168652644)

[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](#_Toc168652645)

[「o」「x」が出力された三目並べ画面が出力される。 6](#_Toc168652646)

[2.3 結果表示画面 7](#_Toc168652647)

[勝利画面表示 引き分け画面表示 7](#_Toc168652648)

[2.4 終了選択画面 7](#_Toc168652649)

[もう一度遊ぶ…0を選んだ場合 7](#_Toc168652650)

[2.5 エラー出力画面 7](#_Toc168652651)

[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 7](#_Toc168652652)

[既に入力されている箇所を指定した場合のエラー文 8](#_Toc168652653)

[3. ファイル構成 9](#_Toc168652654)

[3.1 tic\_tac\_toe.c 9](#_Toc168652655)

[3.2 get\_input.c 9](#_Toc168652656)

[3.3 get\_output.c 10](#_Toc168652657)

[3.4 get\_judge.c 10](#_Toc168652658)

[3.5 define.h 10](#_Toc168652659)

[ターンを表す列挙型 10](#_Toc168652660)

[真か偽を表す列挙型 10](#_Toc168652661)

[三目並べの続行を表す列挙型 10](#_Toc168652662)

[勝利を表す列挙型 10](#_Toc168652663)

[ターン数の上限を表すマクロ定義 11](#_Toc168652664)

[列数を表すマクロ定義 11](#_Toc168652665)

[行数を表すマクロ定義 11](#_Toc168652666)

[座標を表す構造体 11](#_Toc168652667)

[4. 機能構成 12](#_Toc168652668)

[4.1. tic\_tac\_toe.c 12](#_Toc168652669)

[main()関数 12](#_Toc168652670)

[Mainで使う変数 14](#_Toc168652671)

[4.2. get\_input.c 14](#_Toc168652672)

[InputReset()関数 14](#_Toc168652673)

[InputNumber()関数 14](#_Toc168652674)

[InputMark()関数 15](#_Toc168652675)

[4.3. get\_output.c 15](#_Toc168652676)

[OutputScreen()関数 15](#_Toc168652677)

[OutputTurn ()関数 15](#_Toc168652678)

[OutputResult()関数 16](#_Toc168652679)

[4.4. get\_judge.c 16](#_Toc168652680)

[JudgeTurn()関数 16](#_Toc168652681)

[JudgeRange()関数 16](#_Toc168652682)

[JudgeMark()関数 16](#_Toc168652683)

[JudgeResult()関数 17](#_Toc168652684)

[JudgeRetry()関数 17](#_Toc168652685)

# 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。マス目は縦を「|」（バーティカルバー）で表示させ、横は「-」（マイナス）で表示させ縦と横が交わるところは「+」（プラス）で表示する。（以降、この画面を三目並べ画面と表記する。）

ダイアグラム

自動的に生成された説明ダイアグラム が含まれている画像

自動的に生成された説明

### 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

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。

図形

中程度の精度で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。

図形, 矢印

自動的に生成された説明

### 結果表示画面

勝利画面表示 引き分け画面表示

ダイアグラム, 図形

自動的に生成された説明図形, 矢印

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

図形, 矢印

自動的に生成された説明

### エラー出力画面

##### 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

図形, 矢印

自動的に生成された説明

##### 既に入力されている箇所を指定した場合のエラー文

図形, 矢印

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 | InputReset() | 盤面を初期化 |
| COORD(構造体) | InputNumber() | 座標の入力 |
| void | InputMark() | 「o」「x」のどちらを入力 |

## get\_output.c

|  |  |  |
| --- | --- | --- |
| 型 | 関数名 | 説明 |
| void | OutputScreen() | 三目並べ画面の出力 |
| void | OutputResult() | 結果の出力 |

## get\_judge.c

|  |  |  |
| --- | --- | --- |
| 型 | 関数名 | 説明 |
| TURN(列挙型) | JudgeTurn() | 手番の判定 |
| BOOL(列挙型) | JudgeRange() | 数値の範囲を判定 |
| BOOL(列挙型) | JudgeMark() | 「o」「x」の重複判定 |
| JUDGE(列挙型) | JudgeResult() | 勝敗の判定 |
| EXIT(列挙型) | JudgeRetry() | 続行判定 |

## 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 |

##### ターン数の上限を表すマクロ定義

|  |  |
| --- | --- |
| 型名 | #define TURN\_LIMIT 8 |

##### 列数を表すマクロ定義

|  |  |
| --- | --- |
| 型名 | #define GRIT\_HEIGHT 3 |

##### 行数を表すマクロ定義

|  |  |
| --- | --- |
| 型名 | #define GRIT\_WIDE 3 |

##### 座標を表す構造体

|  |  |  |
| --- | --- | --- |
| 型名 | COORD | |
| メンバ変数 | int x | int y | |

# 機能構成

###### 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

・InputReset()関数、JudgeTurn()関数、OutputScreen関数の順で関数を呼び出し、処理を行う。

・OutputResult()関数を用いて勝敗を表示し、JudgeRetry()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

・InputMark()関数、OutputScreen()関数の順で関数を呼び出し、処理を行う。

・JudgeResult()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OutputTurn()関数、InputNumber ()関数、JudgeRange()関数、JudgeMark()関数の順で関数を呼び出し、処理を行う。

・JudgeRange()関数、JudgeMark()関数いずれかがFALSEの場合、再度ループの処理を行う。どちらも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 |
| 座標の情報を持つ変数 | COORD | なし |
| 盤面の情報を持つ配列の要素 [GRIT\_WIDE][GRIT\_HEIGHT] | char | InputReset関数で初期化 |
| 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 |

###### get\_input.c

InputReset()関数

|  |  |
| --- | --- |
| 引数 | 盤面の情報を持つ配列のポインタ[列数を表すマクロ] (char\*型) |
| 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。 |

InputNumber()関数

|  |  |
| --- | --- |
| 引数 | なし |
| 戻り値 | 座標 (COORD型) |
| 説明 | char型の変数にgetch()を用いて値を入力。 数値化を行いint型の変数に代入。  これを構造体のメンバ変数x,yにそれぞれ代入し、構造体を返す。 |
| 表示画面 | [2.2](#_盤面表示画面) |

InputMark()関数

|  |  |
| --- | --- |
| 引数 | 盤面の情報を持つ配列のポインタ[列数を表すマクロ] (char\*型)  座標の情報を持つ構造体（COORD型）  手番の情報を持つ変数(TURN型) |
| 戻り値 | なし（void型） |
| 説明 | 手番の情報から「o」「x」の入力を場合分けする。 配列のポインタにメンバ変数x,yの値を加算し、ｘ,yが示す配列の要素に「o」「x」を入力する。 |
| 表示画面 | [2.2](#_表示画面) |

###### get\_output.c

OutputScreen()関数

|  |  |
| --- | --- |
| 引数 | 盤面の情報を持つ配列の要素[行数を表すマクロ][列数を表すマクロ](char型) |
| 戻り値 | なし(void型) |
| 説明 | 盤面の情報を持つ配列と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) |

OutputTurn ()関数

|  |  |
| --- | --- |
| 引数 | 手番の情報を持つ変数(int型) |
| 戻り値 | なし(void型) |
| 説明 | 手番の情報をもとにprint文を出力する。 |
| 表示画面 | [2.2](#_表示画面) |

OutputResult()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 | 勝敗の情報をもとに結果を出力する。 | |
| 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) |

###### get\_judge.c

JudgeTurn()関数

|  |  |
| --- | --- |
| 引数 | ターン数の情報を持つ変数(int型) |
| 戻り値 | CIRCLEかCROSS(TURN型) |
| 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。 |

JudgeRange()関数

|  |  |
| --- | --- |
| 引数 | 座標の情報を持つ構造体（COORD型） |
| 戻り値 | 真か偽を表す列挙型のTRUEまたはFALSE(BOOL型) |
| 説明 | メンバ変数x,yの値が半角整数値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) |

JudgeMark()関数

|  |  |
| --- | --- |
| 引数 | 盤面の情報を持つ配列の要素[メンバ変数x][メンバ変数y]（char型） |
| 戻り値 | 真か偽を表す列挙型のTRUEまたはFALSE(BOOL型) |
| 説明 | 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) |

JudgeResult()関数

|  |  |
| --- | --- |
| 引数 | 盤面の情報を持つ配列[行数を表すマクロ][列数を表すマクロ](char型)  手番の情報を持つ変数(TURN型)  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(JUDGE型) |
| 説明 | 結果を格納する変数をUNKNOWNで初期化し宣言する。  手番の情報から場合分けをする。  「o」の手番の場合、縦、横、斜めのいずれかで「o」が並んでいる場合O\_WINを結果を格納する変数に代入。「x」が手番の場合も同様に処理を行う。  結果を格納する変数がUNKNOWNのときターン数の情報から引き分け判定の処理を行い、CAPACITY\_FULLの時DROWを結果を格納する変数に代入。  結果を格納した変数を返す。 |

JudgeRetry()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 (EXIT型) |
| 説明 | 「もう一度遊ぶ…０/やめる…０以外」をprintし、0以外を入力するとEND、０を入力するとCONTINUE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) | |