三目並べ詳細仕様書

目次

[**1.** **概要** 1](#_Toc167189793)

[**2.** **用語の定義** 1](#_Toc167189794)

[**3.** **ファイル構成** 2](#_Toc167189795)

[3.1. sanmoku.c(main) 2](#_Toc167189796)

[3.2. board.c 2](#_Toc167189797)

[3.3. board.h 2](#_Toc167189798)

[3.4. input\_num.c 2](#_Toc167189799)

[3.5. input\_num.h 2](#_Toc167189800)

[3.6. decision\_count.c 3](#_Toc167189801)

[3.7. decision\_count.h 3](#_Toc167189802)

[**4.** **画面構成** 4](#_Toc167189803)

[**4.1.** **マス目・〇×の表示** 4](#_Toc167189804)

[**4.2.** **手番の表示** 4](#_Toc167189805)

[**4.3.** **警告文の表示** 4](#_Toc167189806)

[**4.4.** **勝利数とリトライ・終了選択の表示** 5](#_Toc167189807)

[**5.** **機能構成** 6](#_Toc167189808)

[**5.1.** **board.c** 7](#_Toc167189809)

[5.1.1. ShowBoard 7](#_Toc167189810)

[5.1.2. ChangeState 7](#_Toc167189811)

[5.1.3. CheckWin 7](#_Toc167189812)

[5.1.4. InitBoard 7](#_Toc167189813)

[**5.2.** **input\_num.c** 7](#_Toc167189814)

[5.2.1. CheckTurn 8](#_Toc167189815)

[5.2.2. PushNum 8](#_Toc167189816)

[5.2.3. RangeNum 8](#_Toc167189817)

[5.2.4. CheckNum 8](#_Toc167189818)

[5.2.5. ChangeTurn 9](#_Toc167189819)

[**5.3.** **decision\_count.c** 9](#_Toc167189820)

[5.3.1. CountWin 9](#_Toc167189821)

[5.3.2. CheckDraw 9](#_Toc167189822)

[5.3.3. PlayEnd 9](#_Toc167189823)

[5.3.4. PlayRetry 10](#_Toc167189824)

# **概要**

本書は作成する三目並べのファイル構成、機能構成、画面構成を示したものである。

# **用語の定義**

記号 → 〇×のこと

手番 → 先行・後攻の順のこと

マス目 → 3×3 の格子のこと

キー → キーボード上の記号・文字・数字のこと

リトライ → 三目並べをもう1度行うこと

空 → 空というマスの状態を表す文字列が格納されていること

# **ファイル構成**

## sanmoku.c(main)

　勝利か引き分け判定がつくまで、関数を回す処理を入れる。

## board.c

|  |  |  |  |
| --- | --- | --- | --- |
| 関数 | ShowBoard | ChangeState | InitBoard |
| 説明 | 現在の数字や記号の表示 | 入力した値に当てはまるマスの中身の数字と記号を入れ替える | １～９がマスの中に入ったマス目の表示 |

## board.h

## input\_num.c

|  |  |  |  |
| --- | --- | --- | --- |
| 関数 | PushNum | CountWin | PlayEnd |
| 説明 | 値を入力する | 勝利数をカウントする | リトライか終了を入力する |

## input\_num.h

## decision\_count.c

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 関数 | CheckDraw | CheckWin | CheckNum | RangeNum | CheckTurn |
| 説明 | 引き分け判定をする(表示はしない) | 記号が縦・横・斜めに3つ揃っているか判定する | 入力された値の中身に既に記号が入っているかを  判定する | 入力された値が範囲内かを判定する | 現在のターンを判定して表示する |

## decision\_count.h

# **画面構成**

## **マス目・〇×の表示**



マスの数字の左右は半角スペース2個の幅、数字の上下はスペース１個分の高さ、マスの数字は全角で表示する。

マス目を表示し、入力された値のマスに〇×を表示する。

マスのスペースは

## **手番の表示**



4.1.の図の下に左の図にある手番の文章が表示される。



## **警告文の表示**



１～９の値以外が入力された場合に表示される。



記号の重複判定に当てはまった場合に表示される。

リトライ・終了の選択でTとQ以外が入力された場合に表示される。



## **勝利数とリトライ・終了選択の表示**



勝ち判定された場合勝利数を4.1の図の下に表示する。

その後勝利数の下にリトライ・終了選択文が表示される。



# **機能構成**

フローチャート  
ダイアグラム

自動的に生成された説明

## **board.c**

### ShowBoard

関数の型 ：void

引数 ：なし

戻り値 ：なし

マス目の中身に数字または記号が入っているか判定。

### ChangeState

関数の型 ：int

引数 ：入力した値、ターン数

戻り値 ：入力された値に該当するマスに現在のターンの記号を入れる

入力された値に当てはまるマスの中身に入っている数字を現在のターンの記号に置き換えて、置き換えたマスの中身を返す。

### CheckWin

関数の型 ：勝利判定をする列挙型

引数 ：なし

戻り値 ：勝利した記号

記号が縦横斜めいずれかに同じ記号が揃っているか判定。

揃っている場合は、揃った記号を返す。揃っていない場合は何も返さない。

### InitBoard

関数の型 ：void

引数 ：マスの配列

戻り値 ：なし

全てのマスの中身に１～９の数字を１つずつ格納する。

## **input\_num.c**

### CheckTurn

関数の型 ： int

引数 ：ターン数

戻り値 ： 現在どちらのターンか

ループの中でターン数をカウントさせ現在のターン数を受け取り偶数、奇数を判別する。

偶数ならTRUE、奇数ならFALSEを返す。

### PushNum

関数の型 ：char

引数 ：なし

戻り値 ：入力された値

4.2の文章を表示させプレイヤーに入力を求め、入力された値を返す。

### RangeNum

関数の型 ：int

引数 ：入力された値

戻り値 ：入力された値が範囲内であるか範囲外か

入力された値を受け取り、半角の１以上９以下の値か判別する。

範囲内の場合は、TRUEを返す。

範囲外の場合は、FALSEを返す。

### CheckNum

関数の型 ：int

引数 ：入力された値

戻り値 ：重複しているか、していないか

入力された値を受け取り、値に該当するマスの中身に記号が格納されているか判別する。

格納されていない場合は、TRUEを返す。

格納されている場合は、FALSEを返す。

## **decision\_count.c**

### CountWin

関数の型 ：int

引数 ：勝利した記号

戻り値 ：勝利数

勝利した記号を受け取り、該当する記号を判別する。

〇が勝利した場合、〇の勝利数を１加算して返す。

×が勝利した場合、×の勝利数を１加算して返す。

### CheckDraw

関数の型 ：int

引数 ：main関数でカウントしたターン数

戻り値 ：三目並べ終了、三目並べ継続

Mainのループの中でターン数をカウントさせ現在のターン数を受け取り、全てのマスが記号で埋まっているか判別する。

全てのマスが埋まった場合、三目並べ終了を返す。

全てのマスが埋まっていない場合、三目並べ継続を返す。

### PlayEnd

関数の型 ：リトライか終了か表す列挙型

引数 ：三目並べ終了

戻り値 ：リトライか終了

三目並べ終了を受け取り、4.4の文章を表示させプレイヤーに入力を求め、リトライか終了かのどちらかが入力されたか判別する。

リトライの場合は、リトライを返す。

終了の場合は、終了を返す。

## define.h