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 1](#_Toc166761376)

[2. 画面遷移図 1](#_Toc166761377)

[3. システム機能フロー図 2](#_Toc166761378)

[3.1 フロー説明 2](#_Toc166761379)

[3.1.1 [プレイヤー名を入力] 2](#_Toc166761380)

[3.1.2 [プレイヤー名は10文字以内ですか] 2](#_Toc166761381)

[3.1.3 [盤面を初期化] 2](#_Toc166761382)

[3.1.4 [先攻後攻をランダムで決定] 3](#_Toc166761383)

[3.1.5 [3×3のマスとプレイヤー名と先後を画面出力] 3](#_Toc166761384)

[3.1.6 [縦横のマス数を入力] 3](#_Toc166761385)

[3.1.7 [入力した値は規定値内か] 4](#_Toc166761386)

[3.1.8 [既に入力されていないか] 4](#_Toc166761387)

[3.1.9 [三目並べの画面のマスに「〇」「×」を出力] 4](#_Toc166761388)

[3.1.10 [一直線がすべて「〇」、または「×」ですか] 4](#_Toc166761389)

[3.1.11 [出力したのは9回目ですか] 5](#_Toc166761390)

[3.1.12 [勝ったプレイヤー名を出力] 5](#_Toc166761391)

[3.1.13 [引き分けを出力] 5](#_Toc166761392)

[3.1.14 [もう一度遊びますか] 5](#_Toc166761393)

[4. ファイル構成図 6](#_Toc166761394)

[4.1 ファイル構成 6](#_Toc166761395)

[4.1.1. tic\_tac\_toe.c 6](#_Toc166761396)

[4.1.2. game\_ready.c 6](#_Toc166761397)

[4.1.3. get\_input.c 8](#_Toc166761398)

[4.1.4. get\_output.c 9](#_Toc166761399)

[4.1.5. get\_result.c 10](#_Toc166761400)

# 概要

本詳細仕様書は、二人のプレイヤーが交互に縦横それぞれ1～3の数字を入力し、マスに「〇」または「×」を出力する三目並べの詳細を示すものである。

# 画面遷移図

ダイアグラム

自動的に生成された説明

# システム機能フロー図

ダイアグラム

自動的に生成された説明

### フロー説明

#### [プレイヤー名を入力]

プレイヤー名をtic\_tac\_toe.cで入力する。

#### [プレイヤー名は10文字以内ですか]

入力したプレイヤー名が10文字以上であれば10文字以内で入力してくださいを出力し[3.1.1](#_[プレイヤー名を入力])に戻る。また10文字以内であれば[3.1.3](#_[盤面を初期化])に移行する。

#### [盤面を初期化]

3×3のマスをすべて空白にし、[3.1.4](#_[先攻後攻をランダムで決定])に移行する。

#### [先攻後攻をランダムで決定]

ランダムで1P先攻後攻が決定され、それに基づいて2Pの先攻後攻を決定する。また先攻が「〇」、後攻が「×」とする。

#### [3×3のマスとプレイヤー名と先後を画面出力]

「〇」「×」を出力するための3×3のマスと、[3.1.1](#_[プレイヤー名を入力])で入力したプレイヤー名、[3.1.4](#_[先攻後攻をランダムで決定])で決定された先攻後攻と「〇」「×」が出力し、[3.1.6](#_[縦横のマス数を入力])へ移行する。

ダイアグラム

中程度の精度で自動的に生成された説明

#### [縦横のマス数を入力]

「〇」「×」を出力した回数をもとに、（1P or 2P）の番です。と出力し

縦の値を半角で1～3の整数値で入力してください。

横の値を半角で1～3の整数値で入力してください。を出力し、プレイヤーが数値を入力する。

#### [入力した値は規定値内か]

[3.1.6](#_[縦横のマス数を入力])で入力した数値が1～3の内の半角整数値であるかを判別し、1～3以内の半角整数値でなければ半角で1～3の以内の整数値を入力してください。を出力し、[3.1.6](#_[縦横のマス数を入力])に移行し、1～3以内であれば[3.1.8](#_[既に入力されていないか])に移行する。

#### [既に入力されていないか]

入力した数値の位置が既に埋まっていないかを判別し、入力されているなら入力済みのマスです。他のマスを選んでください。を出力し[3.1.6](#_[縦横のマス数を入力])に移行し、入力されてなければ[3.1.9](#_[「〇」「×」を交互に出力])に移行する。

#### [三目並べの画面のマスに「〇」「×」を出力]

三目並べの画面のマスに「〇」「×」を出力した回数をもとに、偶数回数は「〇」、奇数回数は「×」を出力し、[3.1.10](#_[一直線がすべて「〇」、または「×」ですか])に移行する。

ダイアグラム

中程度の精度で自動的に生成された説明

#### [一直線がすべて「〇」、または「×」ですか]

縦、横、斜めのいずれかに三つの「〇」もしくは「×」が並んでいるかを判別し、並んでいなければ[3.1.11](#_[出力したのは9回目ですか])に移行し、並んでいれば[3.1.12](#_[勝ったプレイヤー名を出力])に移行する。

#### [出力したのは9回目ですか]

「〇」と「×」が3×3のマス全てが埋まってあるかを判別し埋まってなければ[3.1.6](#_[縦横のマス数を入力])に移行し、埋まっていれば[3.1.13](#_[引き分けを出力]_1)に移行する。

#### [勝ったプレイヤー名を出力]

「勝利したプレイヤー名」が勝ちました！を出力し、[3.1.14](#_[もう一度遊びますか])に移行する。

#### [引き分けを出力]

引き分けでした。を出力し、[3.1.14](#_[もう一度遊びますか])に移行する。

#### [もう一度遊びますか]

もう一度遊ぶ…0／やめる…0以外を出力する。

0を入力で「もう一度遊ぶ」、0以外を入力で「やめる」を選択する。

0を選択すると、[3.1.3](#_[盤面を初期化])に移行する。

0以外を選択すると、ゲームを終わる。

# ファイル構成図

テーブル

自動的に生成された説明

## ファイル構成

## 機能構成

##### tic\_tac\_toe.c

main関数を記述する

プレイヤー名入力についてのループ

プレイヤー名をそれぞれ入力する。

ループ内では全角半角スペース問わず１０文字以内での入力であるかNameRange()関数を用いて判定しTRUEの時終了する。

InitSquare()関数、PreScreen()関数の順で関数を呼び出し、処理を行う。

縦横の入力についてのループ  
入力の判定を通るまで回すループ

ループ内ではSquareNumber ()関数、Numrange()関数、MarkChecker()関数の順で関数を呼び出し、処理を行う。

Numrange()関数、MarkChecker()どちらもTRUEの時ループを終了する。

〇×出力に関するループ

全てのマスに「〇」「×」が出力されるもしくは縦横斜めで３つ「〇」「×」が出力されるまで回すループ

ループ回数をカウントし偶数回「〇」奇数回「×」を出力するように切り替える。  
PrintMark()関数で「〇」「×」を含んだ三目並べ画面を出力する。

Output\_count()関数がFULL、またはBingo()関数がBINGOのときループが終了する。

ゲームを終了するループ

GameResult()関数を用いて勝敗を表示し、PlayAgain()関数を用いてENDの時はゲームを終了しAGAINの時はプレイヤー名入力ループから同様に始まる。

##### game\_ready.c

NameRange()関数  
引数：プレイヤー名 戻り値：TRUEまたはFALSE  
説明：10文字以内であるかを判別し、10文字以下であれば戻り値はTRUEとなり、11文字以上であれば警告文を出力し、戻り値はFALSEとなる。

InitSquare()関数  
引数：〇×を入力するための二次元配列 戻り値：空の二次元配列  
説明：配列を初期化する。

PreScreen()関数  
引数：二次元配列、プレイヤー名 戻り値：なし  
説明：「〇」「×」を出力するための3×3のマスとそのマスごとに空の二次元配列、入力したプレイヤー名、先攻後攻と「〇」「×」が出力する。

##### get\_input.c

SquareNumber ()関数  
引数：要素数２の一次元配列のポインタ 戻り値：なし  
説明：縦と横のマス数を配列に入力する。

Numrange()関数  
引数：要素数２の一次元配列　戻り値：TRUEまたはFALSE  
説明：配列に入力した数字が半角整数値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。

MarkChecker()関数  
引数： 戻り値：TRUEまたはFALSE  
説明：プレイヤーが指定した配列の要素番号に「〇」「×」が既に入力されてあるときはFALSE、またされてないときはTRUEを返す。

##### get\_output.c

PrintMark()関数  
引数： 戻り値：  
説明：

Bingo()関数  
引数：〇×が入力された二次元配列のポインタ 戻り値：TRUEまたはFALSE  
説明：「〇」「×」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TRUE、並んでいないときはFALSEを返す。

SquareFull()関数  
引数：〇×を出力した回数を持つ変数 戻り値：TRUEまたはFALSE  
説明：〇×を出力した回数が

##### get\_result.c

GameResult()関数  
引数： 戻り値：  
説明：

PlayAgain()関数  
引数： 戻り値：  
説明：

##### define.h

TRUE： FALSE： END： AGAIN： FULL： BINGO：