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](#_Toc166751468)

[2. 画面遷移図 1](#_Toc166751469)

[3. システム機能フロー図 2](#_Toc166751470)

[3.1 機能説明 2](#_Toc166751471)

[3.1.1 [プレイヤー名を入力] 2](#_Toc166751472)

[3.1.2 [プレイヤー名は10文字以内ですか] 2](#_Toc166751473)

[3.1.3 [盤面を初期化] 2](#_Toc166751474)

[3.1.4 [先攻後攻をランダムで決定] 3](#_Toc166751475)

[3.1.5 [3×3のマスとプレイヤー名と先後を画面出力] 3](#_Toc166751476)

[3.1.6 [縦横のマス数を入力] 3](#_Toc166751477)

[3.1.7 [入力した値は規定値内か] 4](#_Toc166751478)

[3.1.8 [既に入力されていないか] 4](#_Toc166751479)

[3.1.9 [三目並べの画面のマスに「〇」「×」を出力] 4](#_Toc166751480)

[3.1.10 [一直線がすべて「〇」、または「×」ですか] 4](#_Toc166751481)

[3.1.11 [出力したのは9回目ですか] 5](#_Toc166751482)

[3.1.12 [勝ったプレイヤー名を出力] 5](#_Toc166751483)

[3.1.13 [引き分けを出力] 5](#_Toc166751484)

[3.1.14 [もう一度遊びますか] 5](#_Toc166751485)

[4. ファイル構成図 6](#_Toc166751486)

[5. ファイル構成 7](#_Toc166751487)

[5.1 tic\_tac\_toe.c 7](#_Toc166751488)

[5.2 game\_ready.c 7](#_Toc166751489)

[関数名:InitSquare 7](#_Toc166751490)

[関数名:Context 7](#_Toc166751491)

[関数名:ContextScreen 8](#_Toc166751492)

[関数名SquareScreen 8](#_Toc166751493)

[5.3 get\_input.c 8](#_Toc166751494)

[関数名:SquareNumber 8](#_Toc166751495)

[関数名:NumRange 8](#_Toc166751496)

[関数名:MarkChecker 9](#_Toc166751497)

[5.4 get\_output.c 9](#_Toc166751498)

[関数名:PrintMark 9](#_Toc166751499)

[関数名:Bingo 9](#_Toc166751500)

[関数名:SquareFull 9](#_Toc166751501)

[5.5 get\_result.c 10](#_Toc166751502)

[関数名:GetWin 10](#_Toc166751503)

[関数名:GetEven 10](#_Toc166751504)

[関数名:PlayAgain 10](#_Toc166751505)

# 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

### 機能説明

#### [プレイヤー名を入力]

プレイヤー名を10文字以内で入力する。

#### [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]

入力されていた「〇」「×」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]

ランダムで先攻後攻が決定されます。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が出力し、[3.1.6](#_[縦横のマス数を入力])へ移行する。

黒い背景に白い文字がある

低い精度で自動的に生成された説明

#### [縦横のマス数を入力]

1Pまたは2Pの番です。と出力し

縦の値を1～3の整数値で入力してください。

横の値を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の内の整数値であるかを判別し、1～3以内の整数値でなければ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」または「×」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。

テキスト

低い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[3.1.5](#_[3×3のマスとプレイヤー名と先後を画面出力])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

## tic\_tac\_toe.c

main関数を記述する。

## game\_ready.c

##### 関数名:InitSquare

引数:なし

戻り値:なし

説明:盤面を初期化する。

関数名:NameScreen

引数:入力したプレイヤー名が10文字以内

戻り値:なし

説明: 10文字以内でプレイヤー名を入力し、0文字以上かつ10文字以内であるかを判別し、プレイヤー名を表示する。

##### 関数名:Context

引数:なし

戻り値:先攻後攻

説明:ランダムに先攻後攻を決定する。

##### 関数名:ContextScreen

引数:先攻後攻

戻り値:なし

説明:先攻後攻を表示する。

##### 関数名SquareScreen

引数:なし

戻り値:なし

説明:三目並べを表示する。

## get\_input.c

##### 関数名:SquareNumber

引数:なし

戻り値:プレイヤーが指定したマスの座標

説明:縦と横のマス数を個別に入力する。

##### 関数名:NumRange

引数:プレイヤーが指定したマスの座標

戻り値:数値が範囲内、または範囲外

説明:入力した数字がマス数の範囲内かどうかを判別する。

##### 関数名:MarkChecker

引数: プレイヤーが指定したマスの座標

戻り値:「〇」「×」が既に入力されてある、またはされてない

説明:「〇」「×」が既に入力されていないかを判別する。

## get\_output.c

##### 関数名:PrintMark

引数:プレイヤーが指定したマスの座標

戻り値: 「〇」「×」を格納した配列

説明:入力されたマス数をもとに、配列に「〇」「×」を格納し、出力する。

##### 関数名:Bingo

引数:「〇」「×」を格納した配列

戻り値:「〇」「×」が三つ並んでいる、または並んでいない

説明:「〇」「×」が縦、横、斜めのどれかに三つ並んだか判別する。

##### 関数名:SquareFull

引数:カウント

戻り値:「〇」と「×」が3×3のマス全て埋まっている、または埋まっていない

説明:3×3のマスがすべて埋まっているか判別する。

## get\_result.c

##### 関数名:GetWin

引数:「〇」「×」が三つ並んでいる

戻り値:三目並べ終了

説明:勝利したプレイヤー名を表示。

##### 関数名:GetEven

引数:「〇」と「×」が3×3のマス全て埋まっている

戻り値:三目並べ終了

説明:引き分けを表示。

##### 関数名:PlayAgain

引数:三目並べ終了

戻り値:もう一度遊ぶ、またはやめる

説明:もう一度遊ぶかやめるかを入力する。