三目並べ詳細仕様書

|  |  |
| --- | --- |
| 作成者 | 佐藤 颯　／　細倉 敦史 |
| 作成日 | 2024/05/17 |

目次

[1. 概要 3](#_Toc166829805)

[2. 用語説明 3](#_Toc166829806)

[3. 画面遷移図例 3](#_Toc166829807)

[4. 画面構成 4](#_Toc166829808)

[5. ファイル構成 4](#_Toc166829809)

[5.1. tic\_tac\_toe.c 4](#_Toc166829810)

[5.2. input.c 4](#_Toc166829811)

[5.3. output.c 5](#_Toc166829812)

[5.4. judge.c 5](#_Toc166829813)

[5.5. insert.c 5](#_Toc166829814)

[5.6. search.c 6](#_Toc166829815)

[5.7. tic\_tac\_toe.h 6](#_Toc166829816)

[5.8. define.h 6](#_Toc166829817)

[6. 機能構成 7](#_Toc166829818)

[6.1. tic\_tac\_toe.cが持つ機能 7](#_Toc166829819)

[6.1.1 勝敗が出るまでループ 7](#_Toc166829820)

[6.1.2 入力に関するループ 7](#_Toc166829821)

[6.1.3 ターンの入れ替え 7](#_Toc166829822)

[6.2. input.cが持つ機能 9](#_Toc166829823)

[6.2.1 Input()関数 9](#_Toc166829824)

[6.3. output.cが持つ機能 10](#_Toc166829825)

[6.3.1 Output\_Info()関数 10](#_Toc166829826)

[6.3.2 Output\_Result()関数 11](#_Toc166829827)

[6.4. judge.cが持つ機能 12](#_Toc166829828)

[6.4.1 Judge\_Grid()関数 12](#_Toc166829829)

[6.4.2 Judge\_Input()関数 13](#_Toc166829830)

[6.4.3 Judge\_Result()関数 14](#_Toc166829831)

[6.5. insert.cが持つ機能 16](#_Toc166829832)

[6.5.1 Insert()関数 16](#_Toc166829833)

[6.6. search.cが持つ機能 16](#_Toc166829834)

[6.6.1 Search()関数 16](#_Toc166829835)

# 概要

本書は、画面遷移例、ファイル構成、機能構成についての図や説明を記載する。

# 用語説明

グリッド：三目並べにおけるマス目のこと。  
勝敗：三目並べにおける、勝ちまたは引き分けといった結果のこと。〇か×を同じ方向に３つ並べられた場合は勝ちとなり、全てのグリッドが埋まっても、〇や×が同じ方向に３つ並べられなかった場合は引き分けとなる。  
図柄：三目並べにおける、〇や×のこと。  
ターン：三目並べにおける、〇の番や×の番のこと。

# 画面遷移図



# 画面構成

盤面の表示は「－」（全角ハイフン）が１１文字、「｜」（全角バーティカルバー）が３文字おきに一つ置くことでグリッドを表示する。グリッドの中は図柄、もしくはグリッド番号の左右に全角でスペースをいれる。表示する図柄、番号は全角で表示する。図柄はO、Xではなく〇、×と表示する。

ダイアグラム, 概略図

自動的に生成された説明ダイアグラム

自動的に生成された説明

# ファイル構成

## tic\_tac\_toe.c

main関数を記述するファイル。関数は以下の通り。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 関数名 | 関数の型 | 引数 | 戻り値 | 動作概要 |
| main | int | void | 0 | メイン |

## input.c

入力部分の関数の定義を記述するファイル。関数は以下の通り。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 関数名 | 関数の型 | 引数 | 戻り値 | 動作概要 |
| Input | char | void | char型変数 | フラグによって、入力を制御する関数。 |

## output.c

出力部分の関数の定義を記述するファイル。関数は以下の通り。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 関数名 | 関数の型 | 引数 | 戻り値 | 動作概要 |
| Output\_Info | void | int型変数、  char型配列[3][3] | void | 盤面とどちらのターンかを表示 する関数。 |
| Output\_Result | void | int型変数、  char型配列[3][3] | void | 盤面と勝敗の結果を表示する 関数。 |

## judge.c

判定部分の関数の定義を記述するファイル。関数は以下の通り。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 関数名 | 関数の型 | 引数 | 戻り値 | 動作概要 |
| Judge\_Result | int | char型配列[3][3]、int型配列[2]、int型変数 | int | 三目並べの勝敗の判定をする関数。 |
| Judge\_Input | int | char | int | 入力が正しいかを判定する 関数。 |
| Judge\_Grid | int | int型配列[2]、char型配列[3][3] | int | グリッドが開いているかを 判定する関数。 |

## insert.c

データの挿入部分の関数の定義を記述するファイル。関数は以下の通り。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 関数名 | 関数の型 | 引数 | 戻り値 | 動作概要 |
| Insert | void | int型配列[2]、char型ポインタ配列[3][3]、  int型変数 | void | 指定されたグリッドに図柄を代入する関数。 |

## search.c

探索部分の関数の定義を記述するファイル。関数は以下の通り。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 関数名 | 関数の型 | 引数 | 戻り値 | 動作概要 |
| Search | int | char型変数、char型配列[3][3] | int型配列[2] | 指定されたグリッドの特定を行う関数。 |

## tic\_tac\_toe.h

三目並べにて使用する関数のプロトタイプ宣言を行っている。  
tic\_tac\_toe.cを除く、上記ソースファイル内の関数のプロトタイプ宣言を行う。

## define.h

以下のマクロ及び構造体を定義している。

|  |  |
| --- | --- |
| マクロ名 | 概要 |
| O\_WIN | 〇が勝ったことを表す |
| X\_WIN | ×が勝ったことを表す |
| DROW | 引き分けを表す |
| CONTINUE | ゲームの続行を表す |
| TRUE | 真 |
| FALSE | 偽 |

また、構造体はtypedefにて定義する。

|  |  |  |
| --- | --- | --- |
| 構造体名 | メンバ変数 | 概要 |
| INDEX | int x, int y | 配列の要素を格納する |

# 機能構成

1. tic\_tac\_toe.cが持つ機能

### 勝敗が出るまでループ

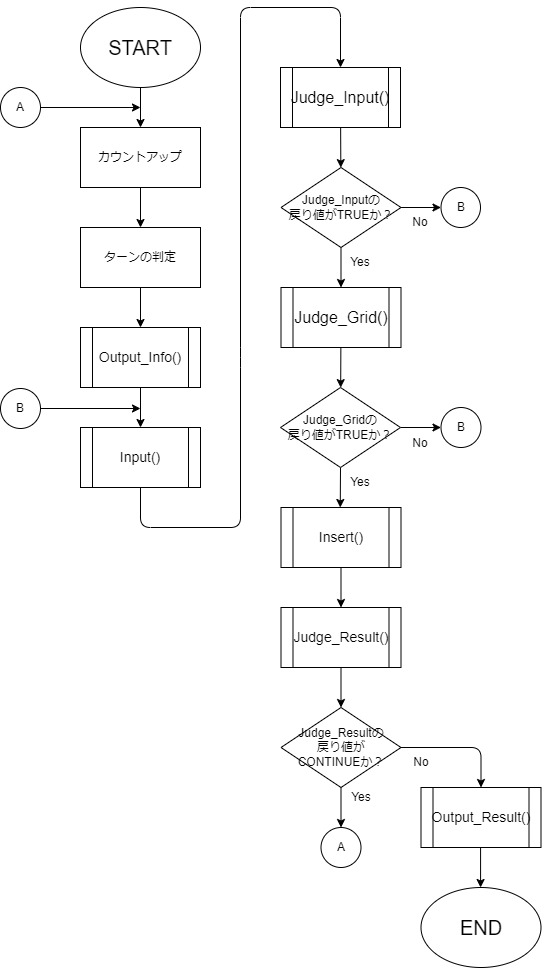
* 勝敗がつくまで回すループ。
* ループ内で他関数の処理をフローチャート通りに行う。
* 勝ちもしくは引き分けの判定を「6.4.3 Judge\_Result()関数」で判断する。

### 入力に関するループ

* 入力の判定が通るまで回すループ。
* ループ内では「6.4.2 Judge\_Input()関数」、「6.6.1 Search()関数」、「6.4.1 Judge\_Grid()関数」の順で関数を呼び出し、処理を行う。
* ループは「6.4.2 Judge\_Input()関数」及び、「6.4.1 Judge\_Grid()関数」の戻り値がどちらもTRUEの場合、ループを抜けるような処理にする。
* 「6.1.1 勝敗が出るまで繰り返しループ」内で行う。

### カウントとターンの判定

* ループ回数をカウントし、カウントが奇数の場合〇、偶数の場合×の番になるようターンを切り替える。
* 「6.1.1 勝敗が出るまで繰り返しループ」内で行う。



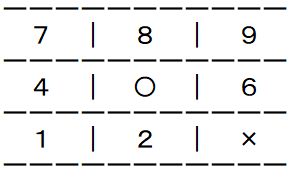
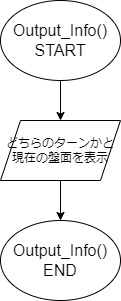
1. input.cが持つ機能
2. Input()関数

* 引数：int-ターンの情報を持つ変数
* 戻り値：char
* 処理内容：キーボードから半角整数１～９のグリッド番号の入力をgetch()を使用して受け付け、char型の変数に格納する。  
  入力前に、入力の内容を伝える文章を表示する。  
  表示する際、現在のターンの図柄を文書内に表示する。  
  〇を置くグリッドを選択してください。  
  入力を戻り値として返す。

テキスト

中程度の精度で自動的に生成された説明

1. output.cが持つ機能
2. Output\_Info()関数

* 引数：int ターンの情報をもつ変数、char 盤面の情報を持つ二次元配列[3][3]
* 戻り値：void
* 処理内容：現在のターンと現在の盤面を表示する。  
  どちらのターンか引数の値から判別し、表示する。  
  ３行３列の配列をうけとり、グリッドとともに配列の内容を表示する。  
  図柄が入っていないグリッドにはそのグリッドの番号を表示する。

1. Output\_Result()関数

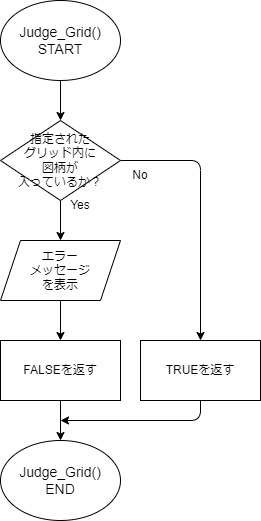
* 引数：char 盤面の情報を持つ二次元配列[3][3]、int 勝敗の情報を持つ変数
* 戻り値：void
* 処理内容：ゲームの勝敗と盤面を表示する。  
  ３行３列の配列をうけとり、グリッドとともに配列の内容を表示する。  
  図柄が入っていないグリッドに橋のグリッドの番号を表示する。  
  どちらのプレイヤーが勝利したか、あるいは引き分けかを勝敗の情報を持つ変数から判定し表示する。

ダイアグラム

自動的に生成された説明

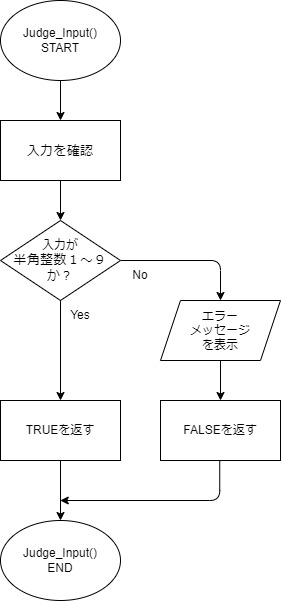
1. judge.cが持つ機能
2. Judge\_Grid()関数

* 引数：INDEX グリッドの要素を指定する構造体、char-盤面の情報を持つ配列[3][3]
* 戻り値：int
* 処理内容：入力を持つ変数により指定されたグリッドの中に、図柄が入っているか確認する。  
  入力されたグリッドの番号に対応する配列の要素を確認し、図柄が配列内に代入されているか確認する。  
  その要素内に、図柄が代入されている場合はエラーメッセージを表示する。  
  空いているグリッドを選択してください。  
  その要素内に、図柄が代入されている場合FALSEを、されていない場合にはTRUEを返す。



1. Judge\_Input()関数

* 引数：char 入力を持つ変数
* 戻り値：int
* 処理内容：1~9の半角整数１文字であるかを判定し、その結果を返す。  
  入力を持つ配列の一つ目の要素の文字コードを確認し、１～９の文字コードを持つかを確認する。  
  半角整数１～９以外の入力が入っていた場合、エラーメッセージを表示する。  
  １～９の半角整数を一つ入力してください。  
  文字列が１～９のみの場合TRUE、それ以外の文字が入っていた場合FALSEを返す。



1. Judge\_Result()関数

* 引数：char 盤面の状態を持つ二次元配列[3][3]、int ターンの情報を持つ変数、INDEX グリッドの要素を指定する構造体
* 戻り値：int
* 処理内容：勝敗が出るか、出る場合どちらが勝ったか、または引き分けかを判定する。グリッドの要素を指定する構造体により、指定されたグリッドの縦、横を見て同じ図柄が３つ並んでいるか確認する。  
  その後、指定されたグリッドにかかわらず、斜めを見て同じ図柄が３つ並んでいるか確認する。  
  同じ図柄が３つ並んでいた場合に、ターンの情報を持つ変数から勝利を表すマクロ（O\_WIN、X\_WIN）を選択し、返す。  
  グリッド内がすべて埋まっており、縦、横、斜めのいずれにも同じ図柄が３つ並んでいない場合、DROWを返す。  
  ゲームの終了が判定されない場合、CONTINUEを返す。

ダイアグラム

自動的に生成された説明

1. insert.cが持つ機能
2. Insert()関数

* 引数：INDEX グリッドの要素を指定する構造体、char 盤面の情報を持つポインタ配列[3][3]、int ターンの情報を持つ変数
* 戻り値：void
* 処理内容：入力された番号のグリッド内に現在のターンの図柄を置く。グリッドの要素を指定する構造体により、指定された配列の要素内に現在のターンの図柄を代入する。  
  図柄の判断にはターンの情報を持つ変数を使用する。

ダイアグラム, 概略図

自動的に生成された説明

1. search.cが持つ機能
2. Search()関数

* 引数：char 入力の情報を持つ変数(1~9)、char 盤面の状態を持つ配列[3][3]
* 戻り値：INDEX
* 処理内容：入力引数の値に対応したグリッドがどの要素なのかを探索し、その要素を返す。  
  グリッド番号１の要素から探索をはじめ、1,２,３,...の順で探索していく。  
  入力引数の番号の数まで探索を続ける。探索した配列の要素を要素数２の配列に格納し、戻り値として返す。

ダイアグラム

自動的に生成された説明