|  |
| --- |
|  |
| 三目並べ詳細仕様書 |
| チーム o |

|  |
| --- |
| 作成者　　遠田 倫也　中嶋　優貴  2024年5月10日 |

[1 概要 1](#_Toc166850051)

[2 用語の定義 1](#_Toc166850052)

[2.1 タイトル画面からの遷移のための用語 1](#_Toc166850053)

[2.1.1 ゲーム終了命令 1](#_Toc166850054)

[2.1.2 ゲーム開始命令 1](#_Toc166850055)

[2.1.3 再入力命令 1](#_Toc166850056)

[2.2 ゲームの終了判定に関する用語 2](#_Toc166850057)

[2.2.1 リザルト表示命令 2](#_Toc166850058)

[2.2.1.1 ゲーム続行命令 2](#_Toc166850059)

[2.2.1.2 リザルト表示命令1 2](#_Toc166850060)

[2.2.1.3 リザルト表示命令2 2](#_Toc166850061)

[2.2.1.4 リザルト表示命令3 2](#_Toc166850062)

[2.3 プレイヤー情報 3](#_Toc166850063)

[2.4 リザルト表示に関する用語 3](#_Toc166850064)

[2.4.1 勝利メッセージ 3](#_Toc166850065)

[2.4.2 引き分けメッセージ 3](#_Toc166850066)

[2.5 ヘッダ部 3](#_Toc166850067)

[2.6 ボディ部 3](#_Toc166850068)

[2.7 入力部 4](#_Toc166850069)

[3 ファイル構成 5](#_Toc166850070)

[4 画面構成 7](#_Toc166850071)

[4.1 ヘッダ部 8](#_Toc166850072)

[4.2 タイトル画面 9](#_Toc166850073)

[4.3 ゲーム画面 10](#_Toc166850074)

[4.4 入力エラー 13](#_Toc166850075)

[4.5 ゲーム結果表示画面 14](#_Toc166850076)

[4.5.1 勝利時表示画面 14](#_Toc166850077)

[4.5.2 引き分け時表示画面 15](#_Toc166850078)

[4.6 終了確認 15](#_Toc166850079)

[5 機能構成 17](#_Toc166850080)

[5.1 マクロ定義 17](#_Toc166850081)

[5.2 列挙型 18](#_Toc166850082)

[5.2.1 ゲーム終了命令 18](#_Toc166850083)

[5.2.2 リザルト表示命令 18](#_Toc166850084)

[5.2.3 BOOL型 18](#_Toc166850085)

[5.3 main関数 20](#_Toc166850086)

[5.4 Play\_Tic\_Tac\_Toe関数 21](#_Toc166850087)

[5.5 isCorrectInput関数 24](#_Toc166850088)

[5.6 Suggset\_Reinput関数 25](#_Toc166850089)

[5.7 isGameStart関数 26](#_Toc166850090)

[5.8 isGameEnded関数 27](#_Toc166850091)

[5.9 Show\_GameTitle関数 28](#_Toc166850092)

[5.10 Show\_ExitGame関数 30](#_Toc166850093)

[5.11 Show\_GameInfo関数 31](#_Toc166850094)

[5.12 Show\_Headder関数 32](#_Toc166850095)

[5.13 Show\_GameResult関数 33](#_Toc166850096)

[5.14 Get\_Input関数 35](#_Toc166850097)

# 概要

本文書は三目並べの詳細仕様に関する記述であり、用語の定義、ファイル構成、画面構成、機能構成を記述したものである。

# 用語の定義

以下は本仕様書における用語の定義である。プレイヤーとは本ゲームを操作する人とする。

## タイトル画面からの遷移のための用語

## ゲーム終了命令

ゲーム終了命令を表す列挙型の列挙子が0の値の時タイトルからゲーム終了確認画面に遷移する。

## ゲーム開始命令

ゲーム終了命令を表す列挙型の列挙子が1の値の時タイトルから三目並べ  
ゲーム処理を開始する。

## 再入力命令

ゲーム終了命令を表す列挙型の列挙子が2の値の時不正な入力が行われたとしてタイトル画面でもう一度入力させる。

## ゲームの終了判定に関する用語

## リザルト表示命令

リザルト表示命令を表す列挙型の列挙子が0、1、2、3の値の時それぞれ、以下のゲーム続行命令、リザルト表示命令１、２、3を表す。

## ゲーム続行命令

リザルト表示命令を表す列挙型の列挙子が0の値の時、リザルト表示をせず、ゲーム続行を表現している。

## リザルト表示命令1

リザルト表示命令を表す列挙型の列挙子が1の値の時、引き分けを表現している。

## リザルト表示命令2

リザルト表示命令を表す列挙型の列挙子が2の値の時、プレイヤー1の勝ちを表現している。

## リザルト表示命令3

リザルト表示命令を表す列挙型の列挙子が3の値の時、プレイヤー2の勝ちを表現している。

## プレイヤー情報

現在どちらのプレイヤーが操作しているのかを表す情報。

## リザルト表示に関する用語

## 勝利メッセージ

以下のメッセージのことである

プレイヤー[現在のプレイヤー番号] ( [現在のプレイヤーの記号] )の勝利

## 引き分けメッセージ

以下のメッセージのことである

引き分け

## ヘッダ部

図1の画面の構造と名称の上部でありゲームのタイトルとゲームバージョンを表示する場所でありすべての画面で同一の表示がされる。

## ボディ部

図1の画面の構造と名称の中央部でありプレイヤーに入力の案内、現在のゲーム状況を表示する場所であり各画面に応じた表示がされる。

## 入力部

図1の画面の構造と名称の下部でありプレイヤーの入力をする場所である。

手紙

自動的に生成された説明

図 1　画面の構造と名称

# ファイル構成

Tic\_Tac\_Toeフォルダおよびソリューション内にmain.c、game.c、display.c、general.cのソースファイルおよび、game.h、display.h、general.hヘッダファイルを作成する。

Tic\_Tac\_Toe ソリューションをまとめたフォルダ

tic\_tac\_toe.sln 三目並べ用のソリューション

main.c 各画面の表示を行う関数及びゲームの処理を行う関数の呼び出しを行うmain関数がある

game.c ゲームの処理とゲーム情報の表示の関数をまとめている。

game.h game.cのヘッダファイル。

display.c 画面表示に関する関数をまとめている。

display.h display.cのヘッダファイル

general.c 各関数で使いまわされる表示や処理に関する関数をまとめている。

general.h general.cのヘッダファイル

列挙型や各種Define、includeの定義をまとめている。

図2は本プログラムの各ファイル間の関係を示している。

ダイアグラム

自動的に生成された説明

図 2 ファイル間の関係

# 画面構成

各節の図番号奇数番目は実装方法を表し、偶数番目はプレイヤーが実際に見る画面である。本章の図中で白枠部分は空白を表しており、数字は挿入する空白の数である。

文字列に関する説明では半角スペース1文字を[]の中に半角スペースの数を入れて、[1]と表す。図3、4は各画面間の遷移を表わす図である。

グラフィカル ユーザー インターフェイス

自動的に生成された説明

図 3　画面の状態遷移図

ダイアグラム

自動的に生成された説明

図 4　画面遷移図

## ヘッダ部

図5、6はヘッダ部画面の構成である。ヘッダ部の左側にゲーム名、右側にゲームバージョンを表示する。表1は画面上に表示されている文字列の実装方法に関する説明である。ヘッダ部は常に表示される。なお表示されている文字列は変更される可能性があるため、三目並べとバージョン表記の間の[54]の空白は文字列を含めて半角76文字分になるよう計算する。

例) 今回の場合

三目並べゲーム = 14文字　ver.1.0 = 7文字　の時

76 – 14 – 7 - 1 = 54文字となる

表 1 ヘッダ画面に表示される文字列の実装方法

|  |  |
| --- | --- |
| 表示文字列 | 実装文字列 |
| 三目並べ | 三目並べ |
| ver1.0 | ver1.0 |

テーブル

自動的に生成された説明

図 5ヘッダ部表示実装方法

テーブル

自動的に生成された説明

図 6　ヘッダ部表示

## タイトル画面

以下の図7、8はタイトル画面の構成である。表2は画面上に表示されている文字列の実装方法に関する説明である。

表 2　タイトル画面に表示される文字列の実装方法

|  |  |
| --- | --- |
| 表示文字列 | 実装文字列 |
| 三目並べゲーム | 三目並べゲーム |
| Press Y to Start Game | Press[1]Y[1]to[1]Start[1]Game |
| Press N to Exit Game | Press[1]N[1]to[1]Exit[2]Game |

|  |  |
| --- | --- |
| テーブル  自動的に生成された説明  図 7　タイトル画面実装方法 | テキスト  自動的に生成された説明  図 8　タイトル画面 |

## ゲーム画面

以下の図9、10はゲーム画面の構成である。

|  |  |
| --- | --- |
| 図 9　ゲーム画面実装方法 | 図 10　ゲーム画面 |

画面左側のマス目は図11のように実装される。

テキスト

中程度の精度で自動的に生成された説明

図 11　マス目の実装方法

図12のようにプレイヤー1が入力すると半角小文字のo、プレイヤー2が入力すると半角小文字のxにマス目の数字が置き換わる。なお内部的にも同様に1~9のインデックスが割り当てられているため、各マス目への参照はこの番号に従って行うものとする。

グラフィカル ユーザー インターフェイス

低い精度で自動的に生成された説明

図 12　ゲーム進行に伴う表示の変化

表3は画面上に表示されている文字列の実装方法に関する説明である。現在の盤面の左右に半角ハイフン5個が配置され、ルールの左右には半角ハイフン17個が配置される。

表 3　ゲーム画面に表示される文字列の実装方法

|  |  |
| --- | --- |
| 表示文字列 | 実装文字列 |
| -----現在の盤面----- | -----現在の盤面----- |
| -----------------ルール----------------- | -----------------ルール----------------- |
| 1. 終了条件 | 1.[1]終了条件 |
| 一列・一行・斜めにoかxが揃うと勝ち | 一列・一行・斜めにoかxが揃うと勝ち |
| 9マス全て揃い勝者がいなければ引き分け | 9マス全て揃い勝者がいなければ引き分け |
| 1. 入力方法 | 2.[1]入力方法 |
| 左のマス目の数字を入力してください | 左のマス目の数字を入力してください |
| oかxが入っているマスは既に入力済みです | oかxが入っているマスは既に入力済みです |

なお入力部には以下の表の通り表示する。以下の表は、プレイヤー1が入力しているときの表示文字列例である。

なおプレイヤー1の「1」の部分は現在のプレイヤーがプレイヤー2の時は2と表示され、また、( o )のところは、プレイヤーが2の時は( x )と表示される。

|  |  |
| --- | --- |
| 表示文字列 | 実装文字列 |
| プレイヤー1( o )の入力 : | プレイヤー%d([1]%c[1])の入力[1] :[1] |

## 入力エラー

以下の図13、14入力エラー画面の構成である。表4は画面上に表示されている文字列の実装方法に関する説明である。

|  |  |
| --- | --- |
| テーブル  自動的に生成された説明  図 13　入力エラーの画面実装方法 | テーブル  自動的に生成された説明  図 14　入力エラー画面 |

表 4 入力エラー画面に表示される文字列の実装方法

|  |  |
| --- | --- |
| 表示文字列 | 実装文字列 |
| 正しいマス目を入力してください | 正しいマス目を入力してください |
| Press Enter to Next Input | Press[1]Enter[1]to[1]Next[1]Input |

## ゲーム結果表示画面

## 勝利時表示画面

以下の図15、16は勝利時画面の構成である。

|  |  |
| --- | --- |
| グラフィカル ユーザー インターフェイス  自動的に生成された説明  図 15　勝利時画面の実装方法 | 図 16　勝利時画面 |

表5は画面上に表示されている文字列の実装方法に関する説明である。oは半角小文字である。なおプレイヤー2が勝利した場合は1の部分が2にoの部分が半角小文字xに変わる。

表 5　勝利時画面に表示される文字列の実装方法

|  |  |
| --- | --- |
| 表示文字列 | 実装文字列 |
| プレイヤー1 ( o ) の勝利 | プレイヤー1[1]([1]o[1])[1]の勝利 |
| Press Enter to Back Title | Press[1]Enter[1]to[1]back[1]Title |

## 引き分け時表示画面

以下の図17、18は引き分け画面の構成である。表6は画面上に表示されている文字列の実装方法に関する説明である。

|  |  |
| --- | --- |
| グラフィカル ユーザー インターフェイス が含まれている画像  自動的に生成された説明  図 17　引き分け画面の実装方法 | 図 18　引き分け画面 |

表 6　引き分け時画面に表示される文字列の実装方法

|  |  |
| --- | --- |
| 表示文字列 | 実装文字列 |
| 引き分け | 引き分け |
| Press Enter to Back Title | Press[1]Enter[1]to[1]back[1]Title |

## 終了確認

以下の図19、20は終了確認画面の構成である。表7は画面上に表示されている文字列の実装方法に関する説明である。

|  |  |
| --- | --- |
| テーブル  中程度の精度で自動的に生成された説明  図 19　終了確認画面の実装方法 | テーブル  自動的に生成された説明  図 20　終了確認画面 |

表 7　終了確認画面に表示される文字列の実装方法

|  |  |
| --- | --- |
| 表示文字列 | 実装文字列 |
| 本当に終了しますか？ | 本当に終了しますか？ |
| はい : Y いいえ : N | はい[1]:[1]Y[3]いいえ[1]:[1]N |

# 機能構成

# マクロ定義

簡単な処理をマクロ定義で行ってもよい。

ゲームのタイトルには今回、「三目並べゲーム」と入る。またバージョン表示は初版は「ver1.0」と入る。

は以下の定義は行うこと。

general.h内で定義

#define タイトルを表す置き換える文字列 “ゲームのタイトル”

#define バージョンを表す置き換える文字列 “現在のバージョン”

#define 枠の横幅を表す置き換える文字列 76

#define プレイヤー1を表す置き換える文字列 1

#define プレイヤー2を表す置き換える文字列 -1

#define ENTERを表す置き換える文字列 ‘\n’

#define Yを表す置き換える文字列 ‘y’

#define Nを表す置き換える文字列 ‘n’

#define プレイヤー1の記号を表す置き換える文字列 ‘o’

#define プレイヤー2の記号を表す置き換える文字列 ‘x’

#define ZHU\_SIZE 3 盤面の珠が並ぶ個数を表す。

# 列挙型

## ゲーム終了命令

0がゲーム終了を表す  
1がゲームの開始を表す

2が不正な入力が行われた時に表す

## リザルト表示命令

0が三目並べゲームの続行を表す

1が三目並べゲームが引き分けで終了したことを表す

2が三目並べゲームがプレイヤー1の勝利で終了したことを表す

3が三目並べゲームがプレイヤー2の勝利で終了したことを表す

## BOOL型

0がFALSE、1がTRUEを表し各種判定結果を表す

# main関数

目的 各画面表示を行う関数及びゲームの処理を行う関数の呼び出し

引数 void

返り値 int型0

変数 ゲーム開始命令/ゲーム終了命令を表す列挙型変数

初期値 : ゲーム終了命令

Show\_ExitGame()の返り値を保存するint型変数

初期値 : 返り値

プログラムの終了を表すBOOL型変数

初期値 : TRUE

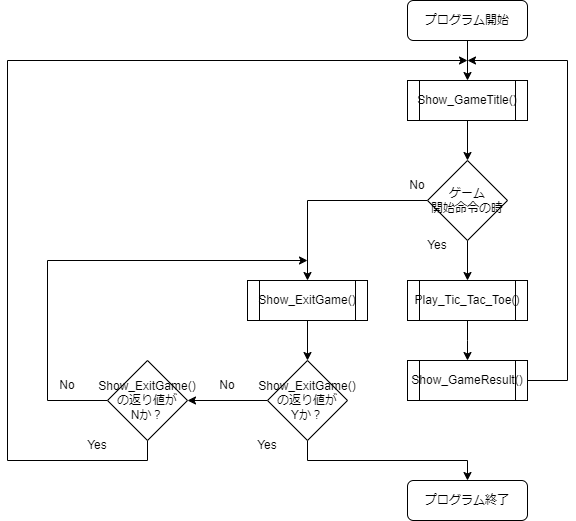


図 21　main関数の処理フローチャート

# Play\_Tic\_Tac\_Toe関数

目的 三目並べのマス表示、ルール表示、ゲーム進行を行う関数。

引数 void

返り値 リザルト表示命令

変数 int型のプレイヤー情報を表す変数

表 8　プレイヤー情報変数のとる値と意味

|  |  |  |
| --- | --- | --- |
|  | 意味 | 値 |
| 初期値 | プレイヤー1 | 1 |
|  | プレイヤー2 | -1 |

現在の盤面の状態を表す任意の型の変数

初期値 : 数字を表す型を利用する場合0を意味する値

　　　 : 文字を意味する型を利用する場合 “”(空白文字)

リザルト表示命令/ゲーム続行命令を表す変数

初期値 : ゲーム続行命令

Get\_Input関数の返り値を受け取るint型変数

初期値 : 0

グラフィカル ユーザー インターフェイス

中程度の精度で自動的に生成された説明

グラフィカル ユーザー インターフェイス

中程度の精度で自動的に生成された説明

図 22　Play\_Tic\_Tac\_Toeの処理フローチャート

# isCorrectInput関数

目的 マス目の指定が正しいかどうかを判定する関数

引数 int型のGet\_Input関数の返り値

返り値 BOOL型のFALSE、TRUE

変数 入力が正しいかの真偽を表す変数

初期値 : FALSE

図形

自動的に生成された説明

図 23　isCorrectInput関数の処理フローチャート

# Suggset\_Reinput関数

目的 マス目の入力が正しくないことを提示してプレイヤーに再入力を求める関数

引数 void

返り値 void

変数 Get\_Input関数の返り値を保存するint型変数

初期値 : 0

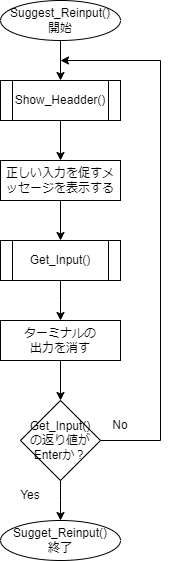


図 24　suggest\_Reinput関数の処理フローチャート

# isGameStart関数

目的 ゲームを開始するか終了するかを判定する関数。

引数 int型のGet\_Input関数の返り値

返り値 列挙型のゲーム開始命令/ゲーム終了命令/再入力命令

変数 ゲーム開始命令/ゲーム終了命令/再入力命令を保存する列挙型変数

初期値 : 再入力命令

ダイアグラム

自動的に生成された説明

図 25 isGameStart関数の処理フローチャート

# isGameEnded関数

目的 ゲーム終了条件を満たして終了するか、ゲームを続行するか判定する関数

引数 int型のプレイヤーを表す変数  
任意の型の盤面を表す変数へのポインタ

返り値 リザルト表示命令

変数 リザルト表示命令/ゲーム続行命令を保存する列挙型の変数

初期値 : ゲーム続行命令

BOOL型　空白が存在することを表す変数。

TRUEならば空白が存在することを表す。

FALSEならば空白が無いことを表す。

初期値 : TRUE

図形 が含まれている画像

自動的に生成された説明

図 26　isGameEnded関数の処理フローチャート

# Show\_GameTitle関数

目的 タイトル画面の表示を行う関数

引数 void

返り値 列挙型 ゲーム終了命令/ゲーム開始命令

変数 列挙型 ゲーム終了命令/ゲーム開始命令を保存する変数

初期値 : 再入力命令

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

図 27 Show\_GameTitle関数の処理フローチャート

# Show\_ExitGame関数

目的 ゲームを終了するかプレイヤーに確認をする画面を表示する関数

引数 void

返り値 int型Get\_Input関数の返り値

変数 Get\_Input関数の返り値を保存するint型変数

初期値 : 0

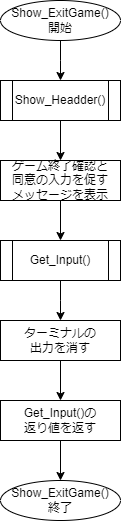


図 28　Show\_ExitGame関数の処理フローチャート

# Show\_GameInfo関数

目的 現在の盤面と三目並べのルールを表示する関数

引数 int型のプレイヤーを表す変数  
 任意の型の盤面を表す変数へのポインタ

返り値 void

テキスト

自動的に生成された説明

図 29　Info関数の処理フローチャート

# Show\_Headder関数

目的 ヘッダ部の情報を表示する関数

引数 char型のゲームタイトルを表す変数でマクロ定義を行う。  
 char型のゲームのバージョンを表す変数でマクロ定義を行う。

返り値 void



図 30　Show\_Headder関数の処理フローチャート

# Show\_GameResult関数

目的 ゲームの勝敗判定に基づき、結果を表示する関数。

引数 リザルト表示命令  
 任意の型の盤面を表す変数へのポインタ

返り値 void

変数 メッセージを保存するchar型変数

初期値 : “”(空文字)

Get\_Input関数の返り値を保存するint型変数

初期値 : 0

グラフィカル ユーザー インターフェイス が含まれている画像

自動的に生成された説明

図 31　 Show\_GameResult関数の処理フローチャート

# Get\_Input関数

目的 入力を1文字キーボードから受け取る関数

引数 void

返り値 int型の\_getch関数の返り値

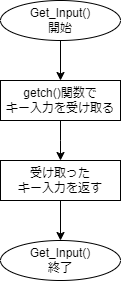


図 32 Get\_Input関数の処理フローチャート