三目並べ\_詳細仕様書

作：高野橋　和

目次

1.　概要

2.　アクティビティ図

3.　関数仕様

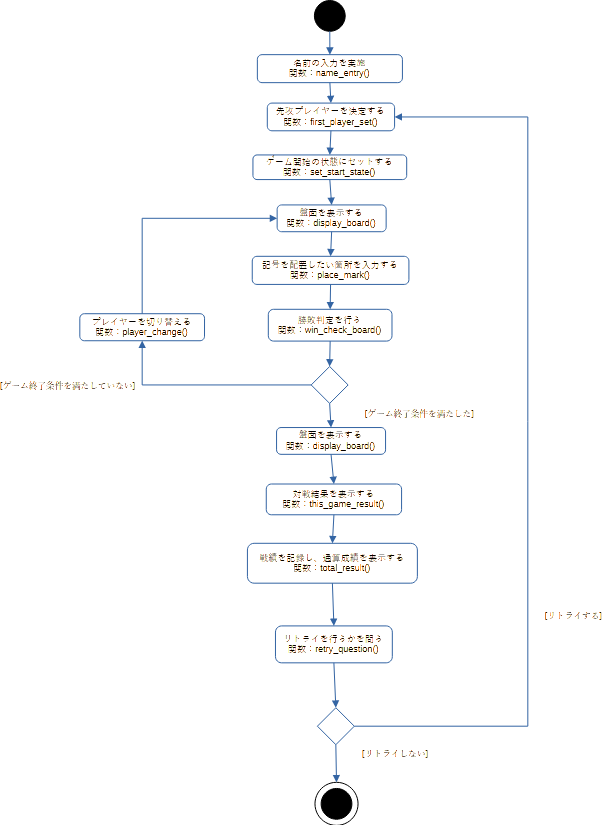
4.　定数仕様

5.　引数定義

1.　概要

　この仕様書では、基本仕様書の「三目並べ仕様書.docx」(金田泰良氏が作成)を基に、三目並べゲームを実装するにあたってのシステムの構成・実装を行う上での詳細な仕様を記していく。

2.　アクティビティ図



3.　関数仕様

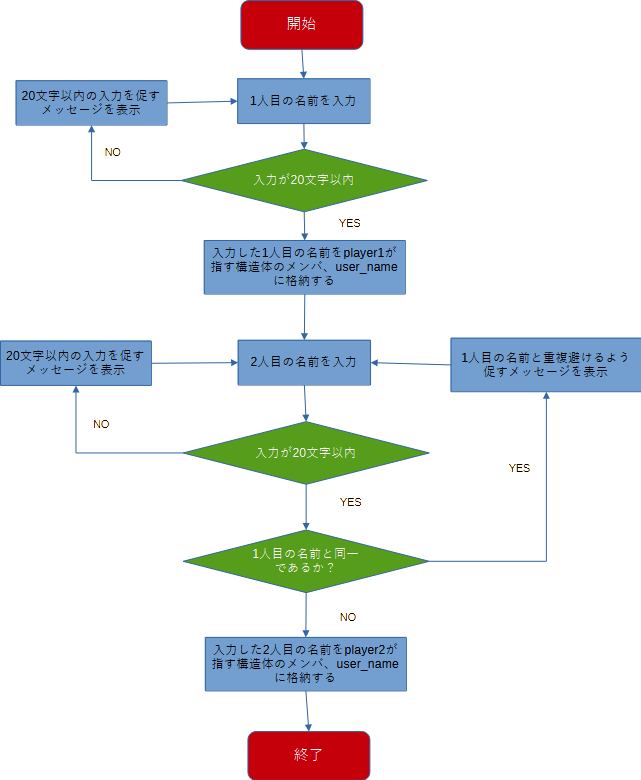
　ここでは、アクティビティ図に登場する機能の順番に記載を行う。

void name\_entry(User\* player1, User\* player2)

　名前の入力を実施するための関数

　この関数で行うことは以下のフローチャートに示す。

フローチャート



備考

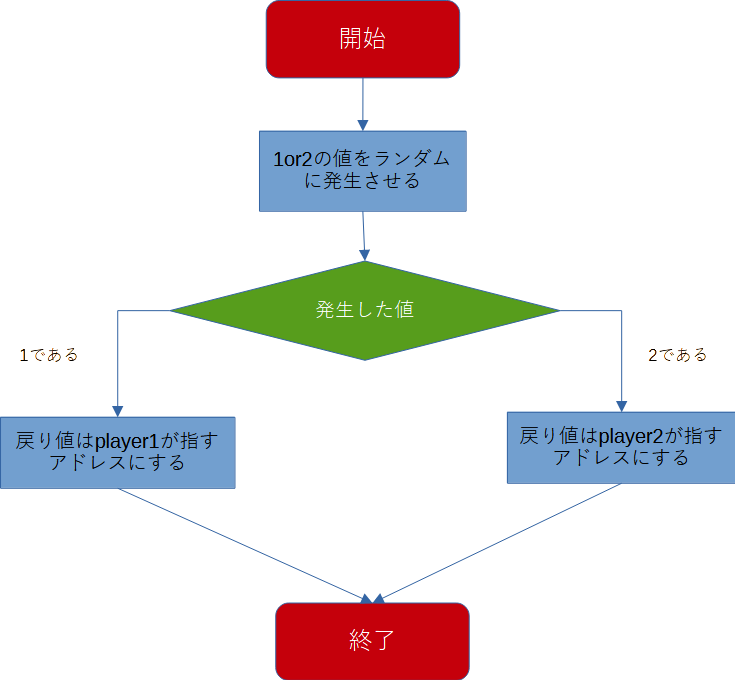
・日本語などのワイド文字を許容する。

User\* first\_player\_set(User\* player1, User\* player2)

　先攻プレイヤーを決定するための関数

　この関数で行うことは以下のフローチャートに示す。

フローチャート



void set\_start\_state(int board[], int\* turn\_count, int\* game\_state)

　ゲーム開始の状態にセットする関数

　この関数で行うことは以下に記す。

・boardが参照する配列に対して、すべての要素（9つを想定）にNO\_PLACE(0)を代入する。

・turn\_countが参照する変数に対して、COUNT\_INIT(1)を代入する。

・game\_state参照する変数に対して、PLAYING\_NOW(0)を代入する。

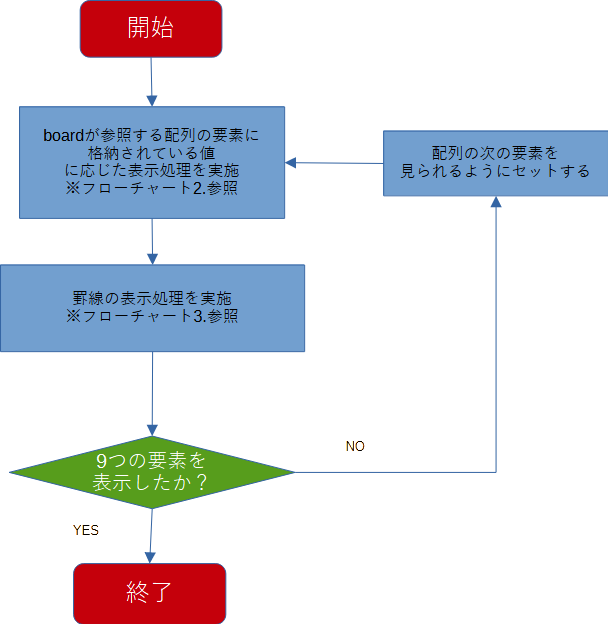
※定数については.、定数定義を参照

void display\_board(int board[])

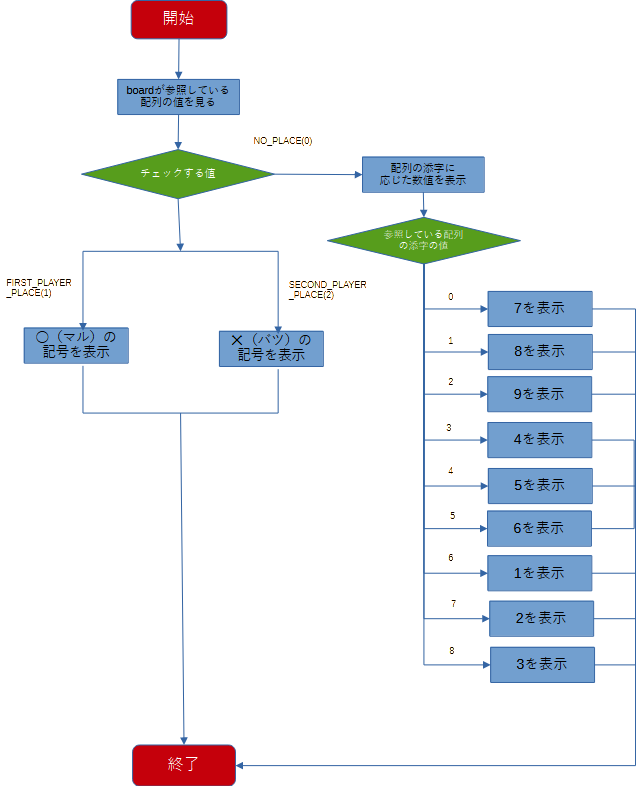
　ゲームの盤面を表示させる関数

　この関数で行うことは以下のフローチャート1~3に示す。

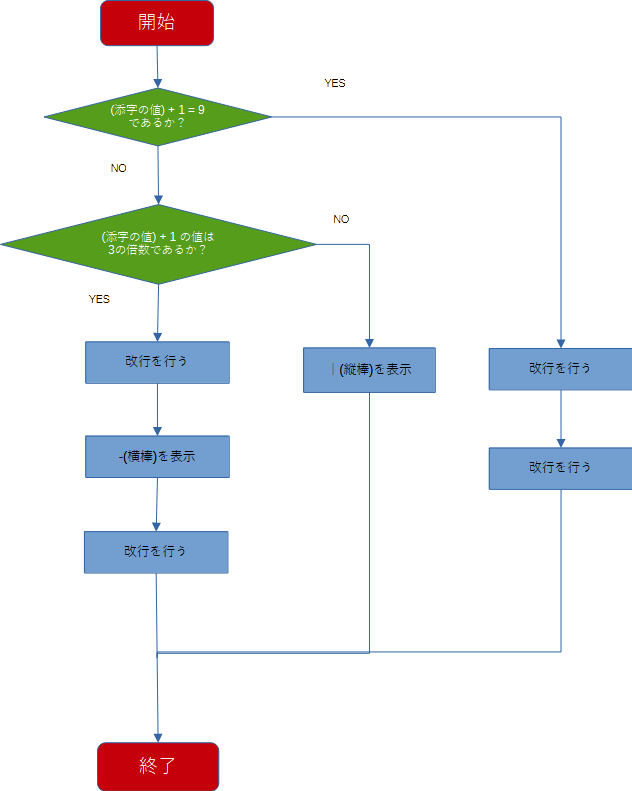
フローチャート1. 関数の流れ



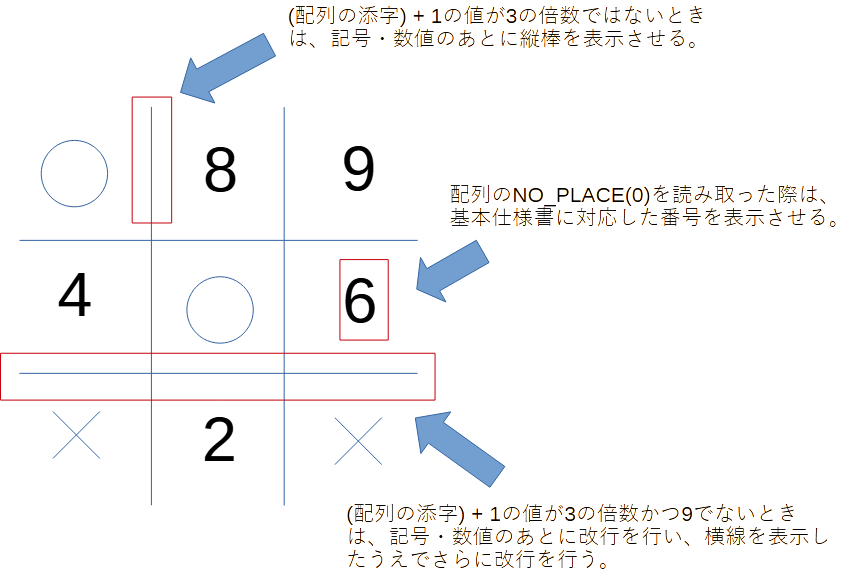
フローチャート2. 配列に格納されている値に応じた表示処理



フローチャート3. 罫線の表示処理



表示処理のイメージ図



備考

・関数呼び出しの際、boardには配列の先頭アドレスを渡すこと。

・○・×・数字・縦棒・横棒について、全角/半角を問わない。数字や記号が混在していても画面表示が崩れないように設計を行うこと。

・NO\_PLACE(0)のときに表示する数字は、以下の通り基本仕様書に定義されているものと連動させている。

例) 配列の[0]の箇所は、 基本仕様書に定義されている7の箇所と対応している。

[0]

[1]

[2]

[4]

[3]

[5]

[8]

[7]

[6]

1

2

3

4

5

6

7

8

9

基本仕様書に定義されている

配置を行う際の盤面に対応する番号

配列の添え字と盤面の対応する箇所

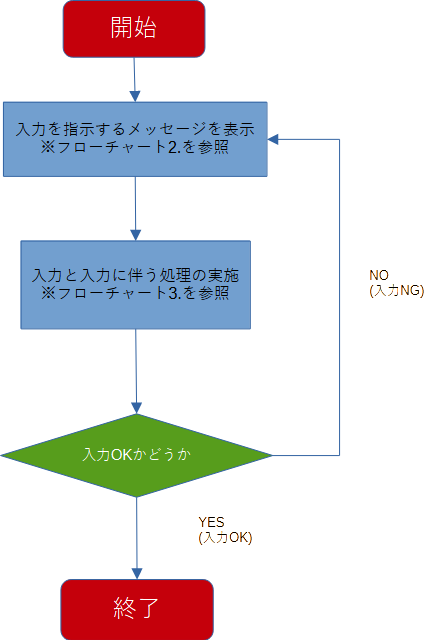
※定数については、定数仕様を参照

void place\_mark(User\* now\_player, int turn\_count, int board[])

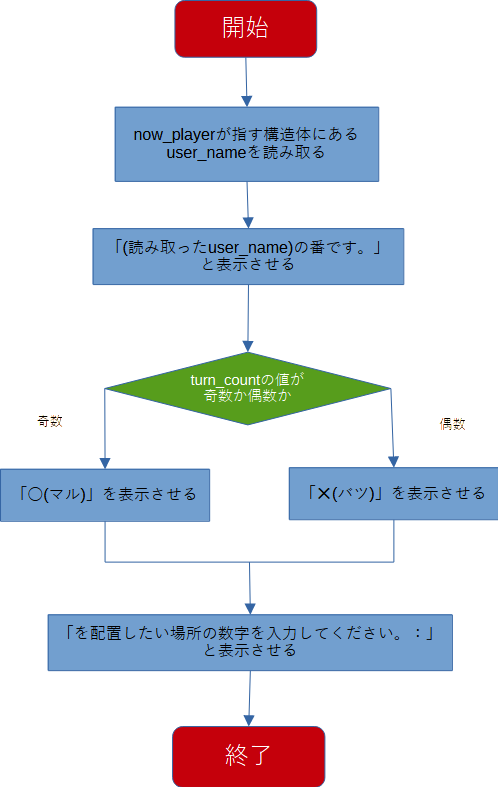
　記号を配置する関数

　この関数で行うことは以下のフローチャート1~3に示す。

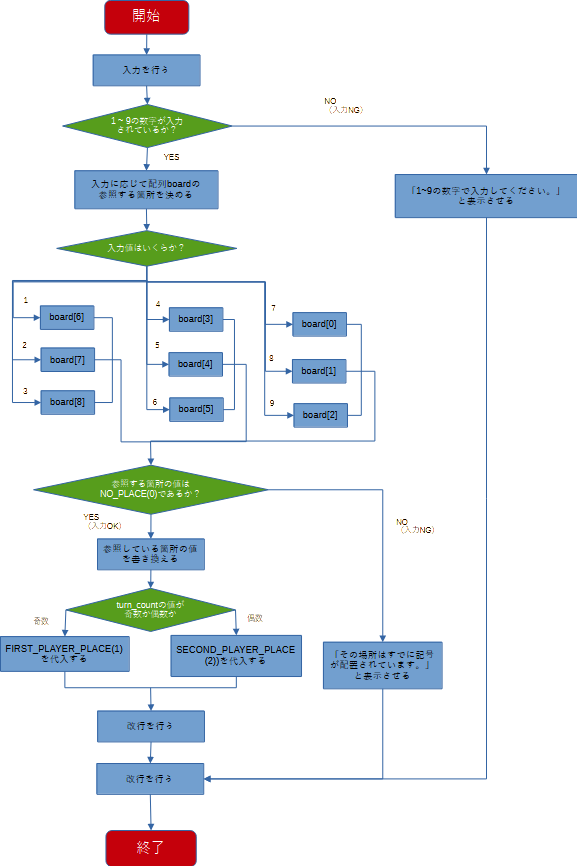
フローチャート1. 関数の流れ



フローチャート2. 入力指示のメッセージを表示



フローチャート3. 入力と入力に対する処理の実施



備考

・関数呼び出しの際、boardには配列の先頭アドレスを渡すこと。

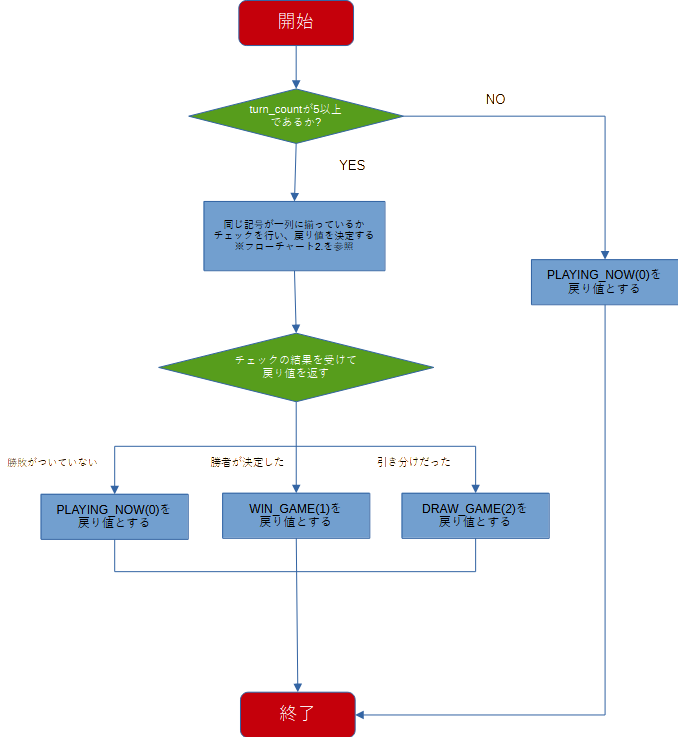
※定数については、定数仕様を参照

int win\_check\_board(int board[], int turn\_count)

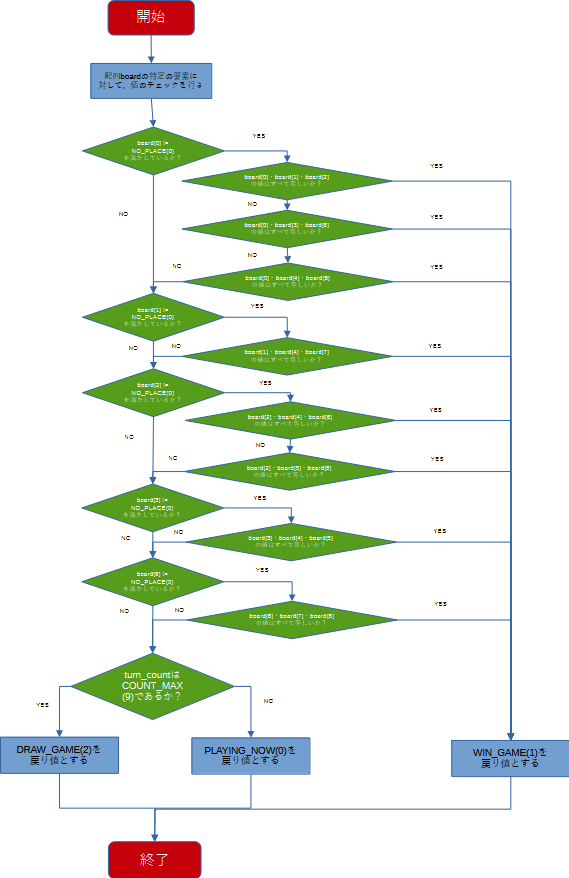
　勝敗判定を行う関数

　この関数で行うことは以下のフローチャート1~2に示す。

フローチャート1. 関数の流れ



フローチャート2.同じ記号が1列に並んでいるかをチェックする



備考

・関数呼び出しの際、boardには配列の先頭アドレスを渡すこと。

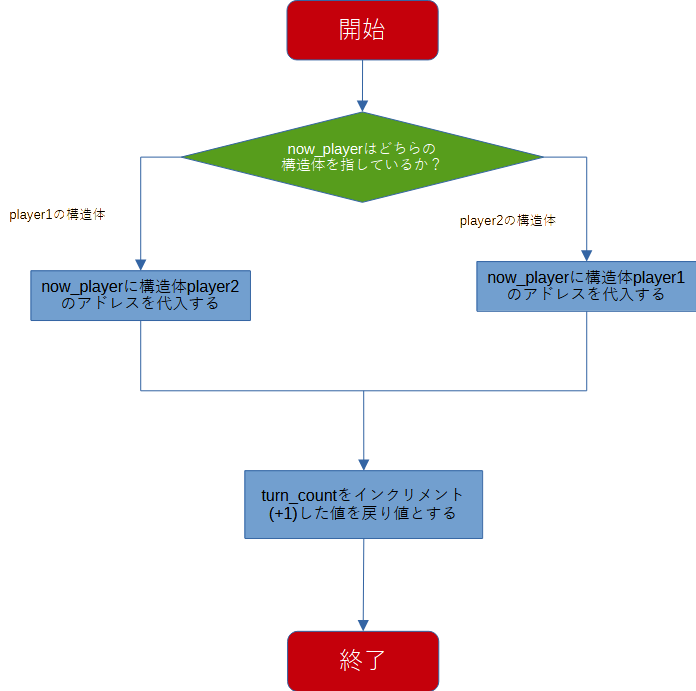
※定数については、定数仕様を参照

int player\_change(User\* player1,User\* player2, User\* now\_player, int turn\_count)

　プレイヤーを切り替える関数

　この関数で行うことは以下のフローチャートに示す。

フローチャート



void this\_game\_result(int game\_state, User\* now\_player)

　対戦結果を表示する関数

　この関数で行うことは以下に記す。

・game\_stateの値がWIN\_GAME(1)の場合、now\_playerが指す構造体にある配列user\_nameを確認し,

「(user\_nameで読み取った名前)が勝利しました。」と表示する。

・game\_stateの値がDRAW\_GAME(2)の場合は、「引き分けです。」と表示する。

・上記のどちらかの表示を終えたら、改行を行う。

※定数については、 定数仕様を参照

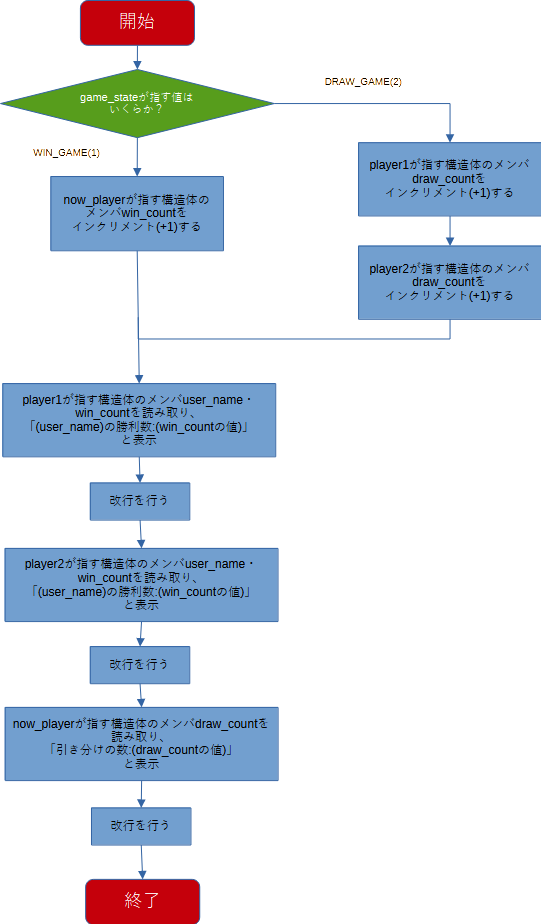
void total\_result(User\* now\_player, User\* player1, User\* player2, int game\_state)

　戦績を記録し、通算成績を表示する関数

　この関数で行うことは以下のフローチャートに示す。

※定数については、定数仕様を参照

フローチャート

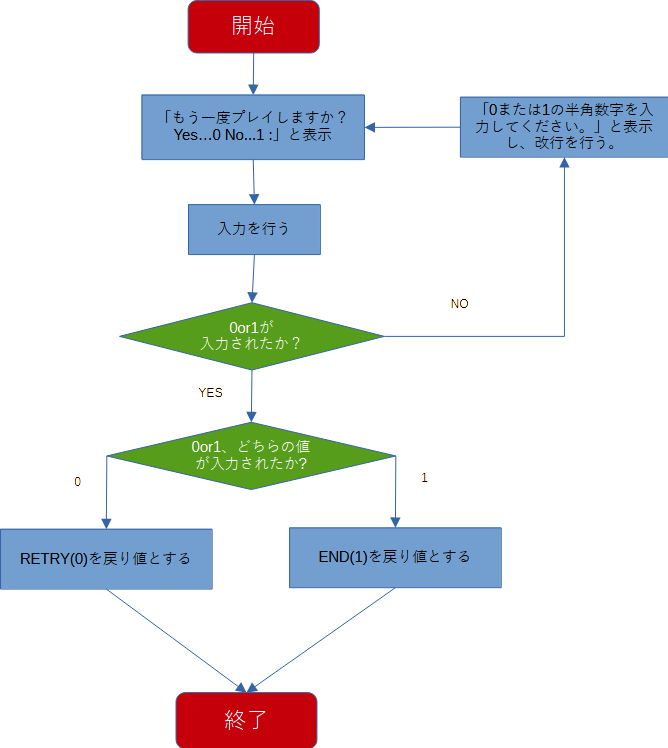


int retry\_question(void)

　リトライを行うかを問う関数

　この関数で行うことは以下のフローチャートに示す。

フローチャート



備考

・RETRY(0)を戻り値とする場合は、first\_player\_set()からの流れをもう一度行うこと。

・END(1)を戻り値とする場合は、この関数終了をもって三目並べゲームを終了させる。

※定数については、定数仕様を参照

4.　定数仕様

　定数マクロとして定義を行うべき定数を記載する。

・構造体Userを定義した際にwin\_count・draw\_countの初期化に用いる定数

ZERO\_INIT(0)・・・戦績がついていないことを指す、初期化の際に用いる

・set\_start\_state()やdisplay\_board()などで使用する引数int board[]で扱う定数

NO\_PLACE(0)・・・記号を配置していないマス、初期化の際に用いる

FIRST\_PLAYER\_PLACE(1)・・・先攻のプレイヤーが記号を配置したマス

SECOND\_PLAYER\_PLACE(2)・・・後攻のプレイヤーが記号を配置したマス

・place\_mark()やwin\_check\_board()などで使用する引数int turn\_countに対して使用する定数

COUNT\_INIT(1)・・・１手目であることを指す初期値、初期化の際に用いる

COUNT\_MAX(9)・・・９手目であることを指す

・win\_check\_board()の戻りとして使用するほか、

　this\_game\_result()やtotal\_result()などで使用する引数int game\_stateに対して使用する定数

PLAYING\_NOW(0)・・・三目並べゲームを行っている最中、初期化の際に用いる

WIN\_GAME(1)・・・ゲームの勝者が決まったことを指す

DRAW\_GAME(2)・・・ゲームの勝敗がつかなかったことを指す

・retry\_check()の戻り値として使用する定数

RETRY(0)・・・もう一度三目並べゲームを行う

END(1)・・・ゲームを終了する

5.　引数定義

　この項目では3.の関数定義で使用する頻度高い引数について定義する。

User型構造体

説明

　名前と戦績を記録するための構造体。2人で遊ぶゲームなので2人分構造体を定義すること。

メンバ

・user\_name(char型配列)・・・プレイヤー名の最大文字数20文字を格納する配列（日本語などワイド文字の入力を認める）。要素数は、日本語の文字のバイト数2 × 最大文字数20 + ヌル文字1で41とする。

・win\_count(int型)・・・勝利数を記録するカウンタ、初期値はZERO\_INIT(0)

・draw\_count(int型)・・・引き分け数を記録するカウンタ、初期値はZERO\_INIT(0)

win\_count・ draw\_countは必ず1ずつ増加させる。

User型のポインタ　\*now\_player

説明

三目並べゲームの手番を行うプレイヤーを指すポインタ変数。

int\*型board[]

説明

　ゲーム中の盤面情報を記録するための配列を指すint型のポインタ。要素数が9である配列をこのポインタに割り当てること。

　配列の参照した際の要素はNO\_PLACE(0)・FIRST\_PLAYER\_PLACE(1)・SECOND\_PLAYER\_PLACE(2)のいずれかの値であるようにすること。

備考

　基本仕様書より入力する際の盤面の番号と、boardに対応させる添字を下記に記す。

　例えば、基本仕様書に定義されている7の箇所は、配列boardの[0]として扱うことを示す。

[0]

[1]

[2]

[4]

[3]

[5]

[8]

[7]

[6]

1

2

3

4

5

6

7

8

9

基本仕様書に定義されている

配置を行う際の盤面の番号

配列game\_boardで

対応させる添字

int型 turn\_count

説明

　先攻後攻合わせて何回手番を行ったかを記録する変数。

　初期値はCOUNT\_INIT(1)、最大値はCOUNT\_MAX(9)とする。

　この変数が奇数である場合は先攻の手番、偶数である場合は後攻の手番を示す。

int型　game\_state

説明

　三目並べゲームの状態を記録する変数。PLAYING\_NOW(0)・WIN\_GAME(1)・DRAW\_GAME(2)

のいずれかの値をとる。