

# 内部仕様書

学生番号： XXXXXX

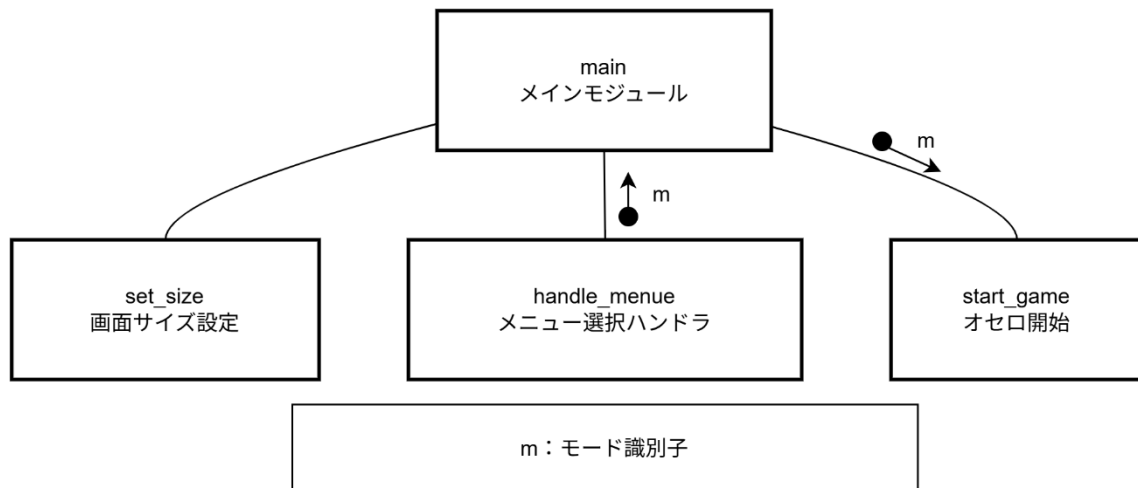
氏 名：Diceke

## ■ 概要

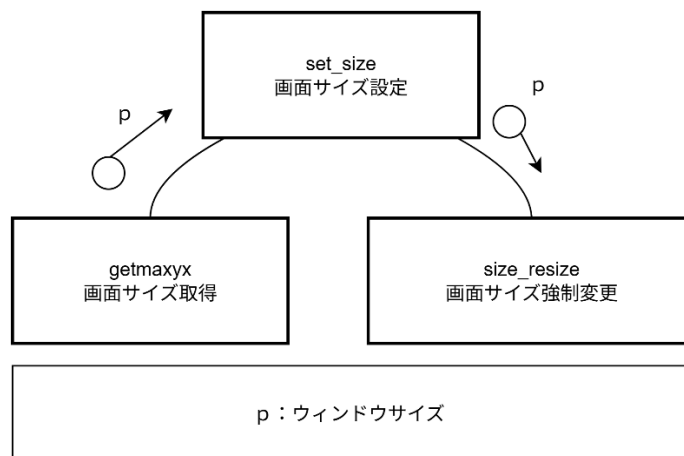
このゲームは 2 人のプレイヤーがそれぞれ黒と白の石を  $8 \times 8$  の盤面に並べ相手の石を挟んで自分の色にすることで自分の石を増やしていき、最終的に盤面にある石の数が多いプレイヤーが勝利するゲームである。自分のターンに相手の石を自分の石でそれぞれ縦、横、斜めに挟むことで、はさんだ相手の石を自分の色に変えることができる。そして、すべてのマスに石が置かれるか、どちらのプレイヤーも打つ手がなくなるとゲームが終了となり、勝敗が決定する。また、ネットワーク対戦機能(同一 LAN 内のみ)、次の手のヒント 機能、CPU 対戦機能がついている。

## ■ モジュール関連図

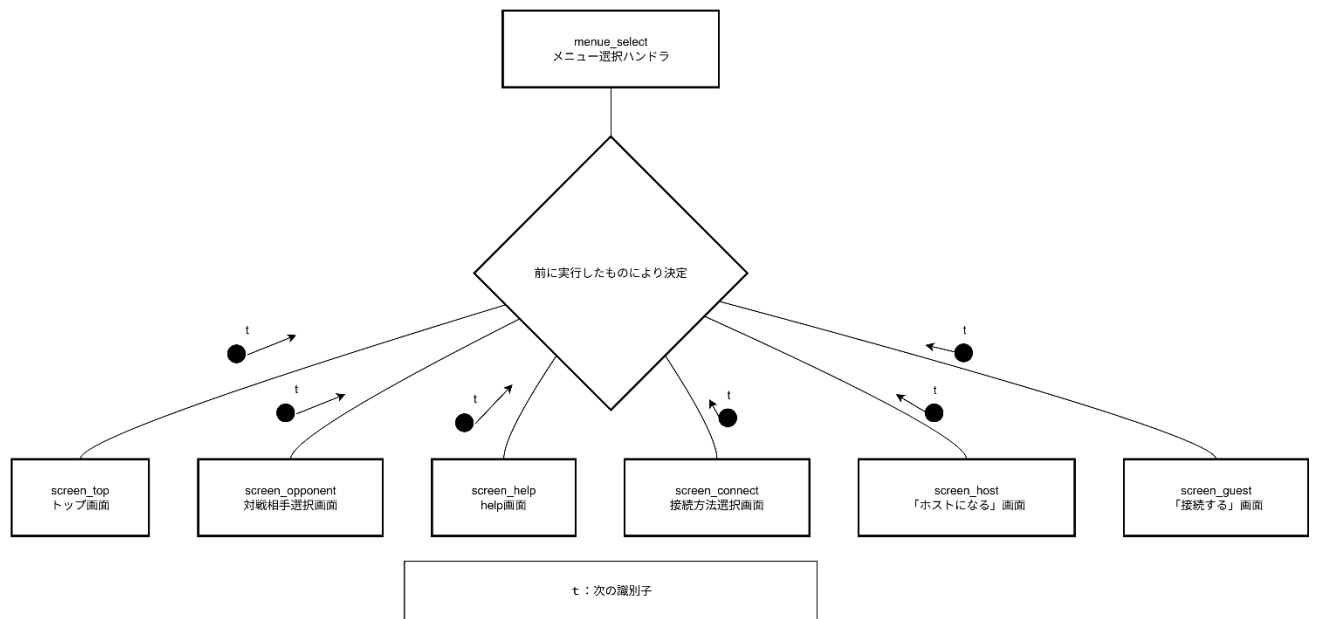
### 1 メインモジュール



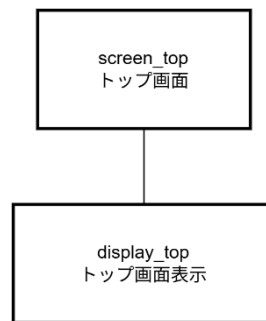
### 1.1 画面サイズ設定モジュール



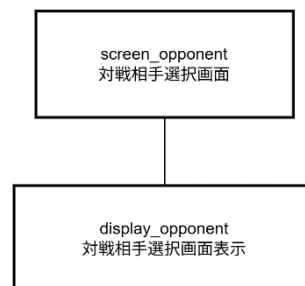
## 1.2 メニュー選択ハンドラ



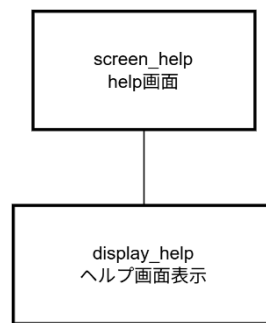
### 1.2.1 トップ画面モジュール



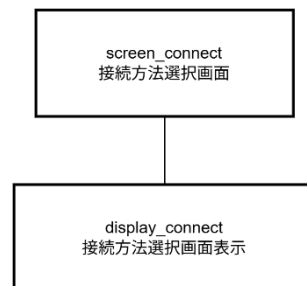
### 1.2.2 対戦相手選択画面モジュール



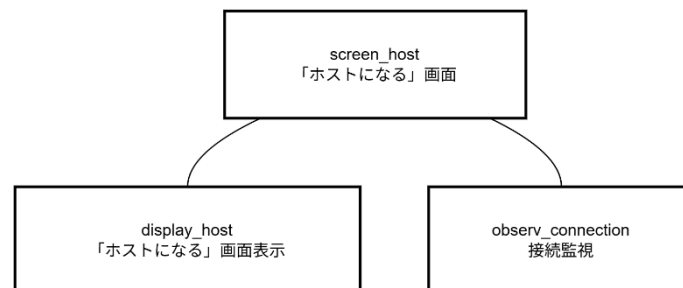
### 1.2.3 help 画面モジュール



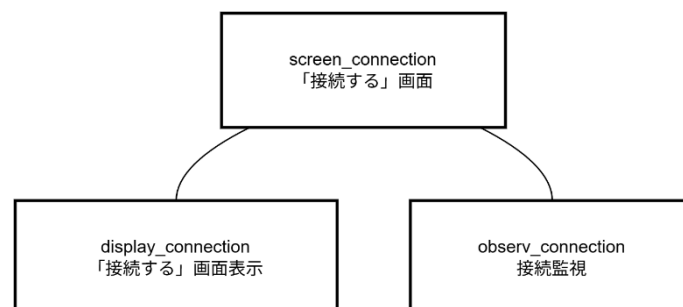
### 1.2.4 接続方法選択画面モジュール



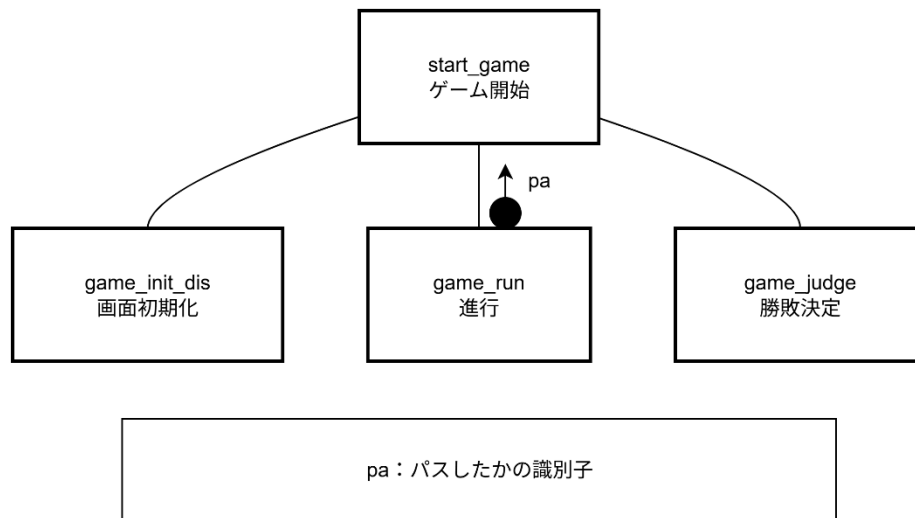
### 1.2.5 「ホストになる」画面モジュール



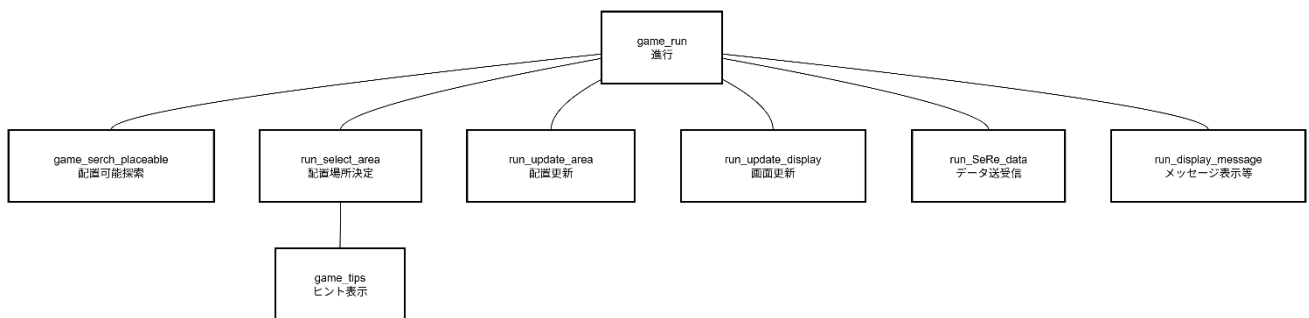
### 1.2.6 「接続する」画面モジュール



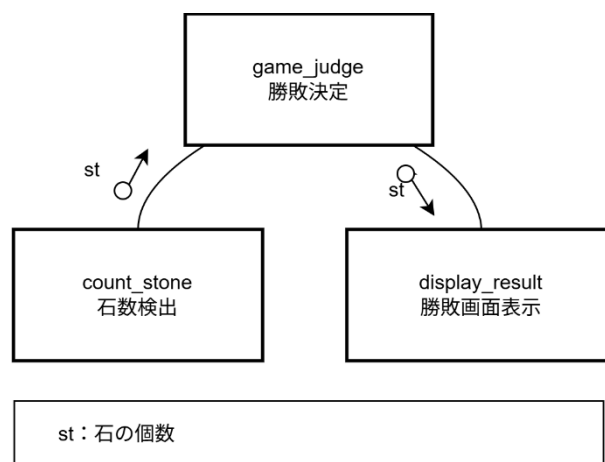
### 1.3 ゲーム開始モジュール



#### 1.3.2 進行モジュール（グローバル変数で管理しているのでデータの移動はありません）

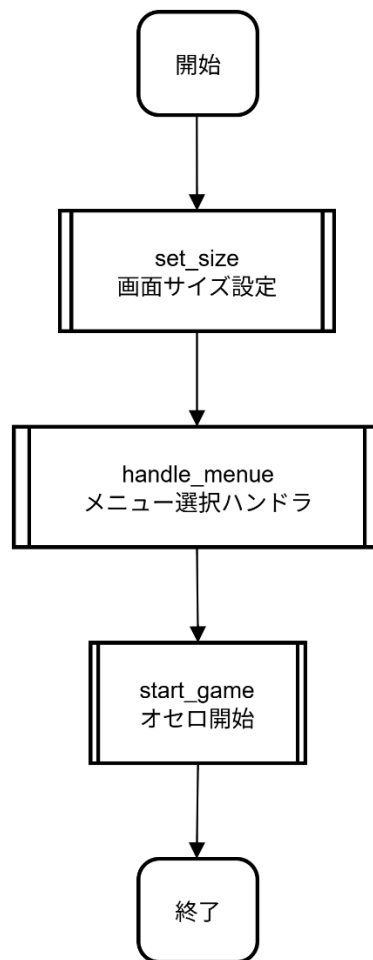


#### 1.3.3 勝敗決定モジュール

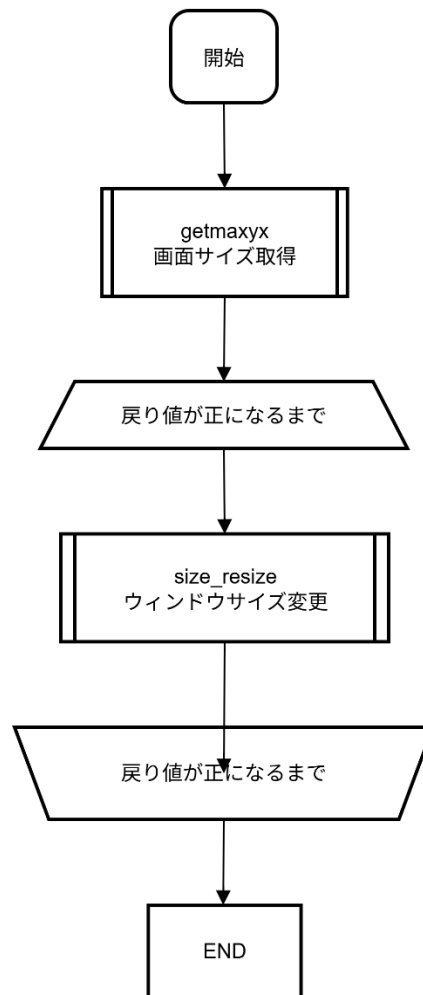


■ モジュールのフロー

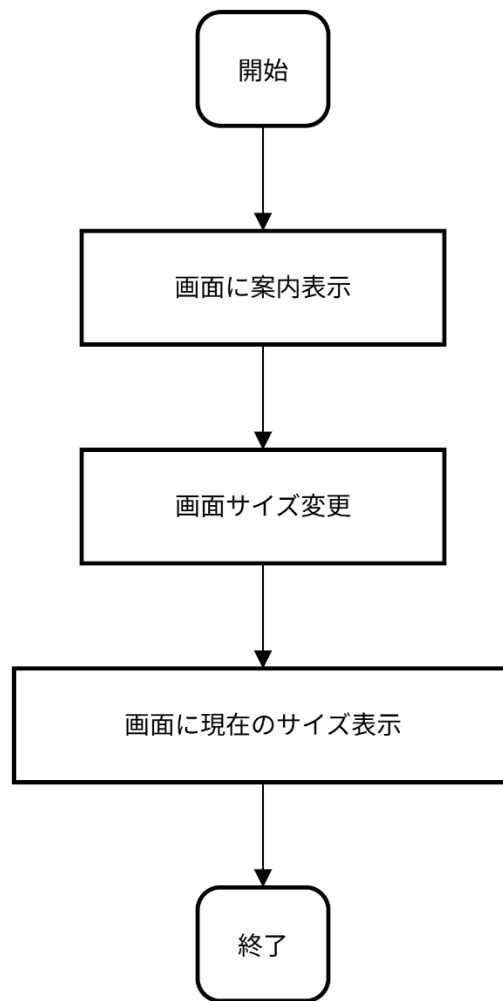
1      メインモジュール



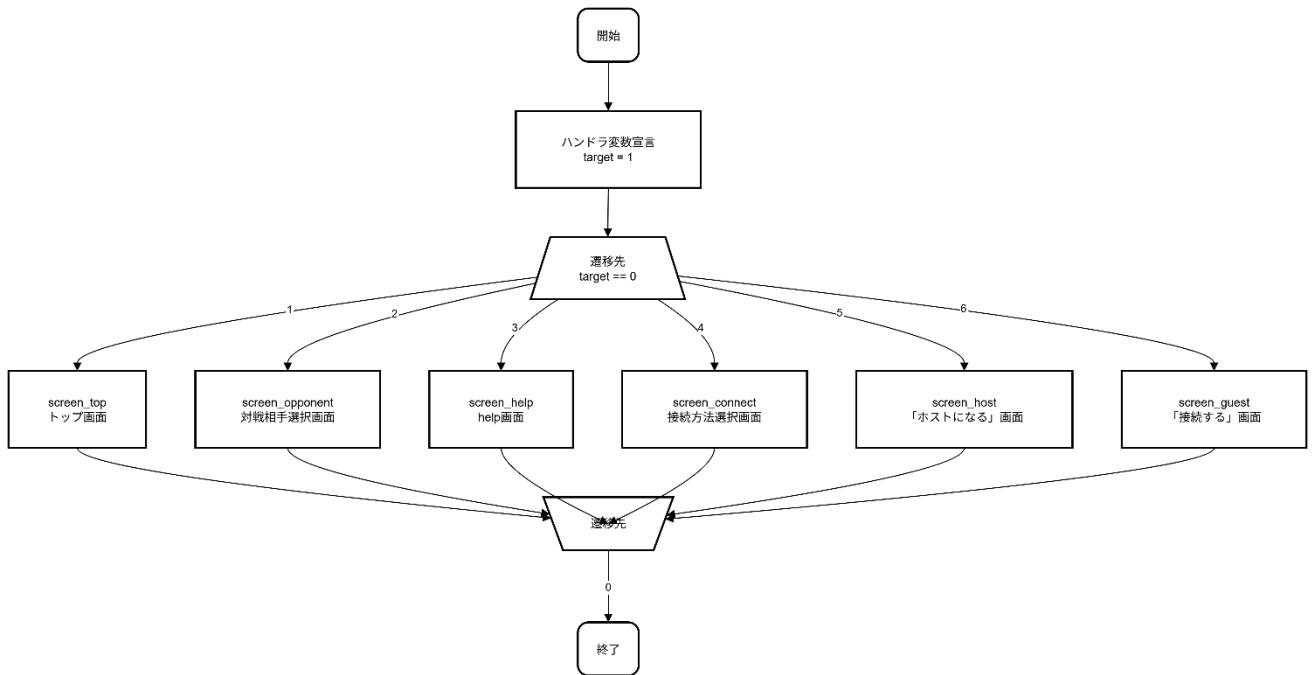
## 1.1 画面サイズ設定モジュール



### 1.1.1 画面サイズ強制変更モジュール

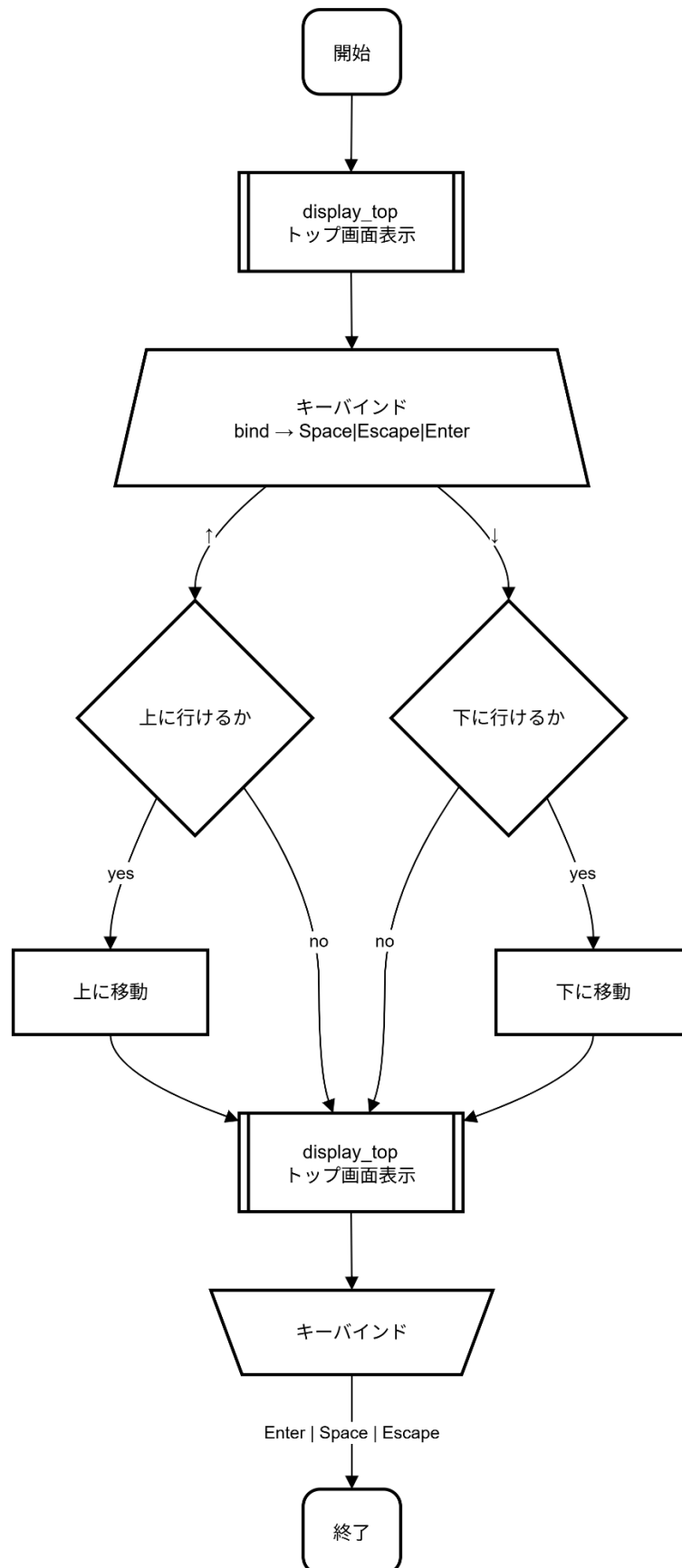


## 1.2 メニュー選択ハンドラ

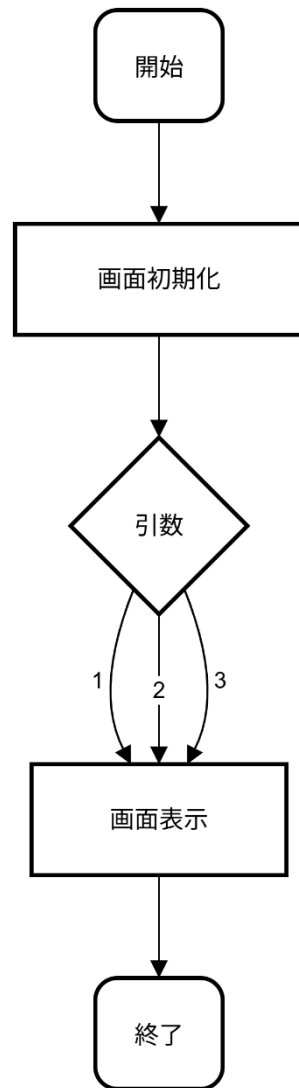




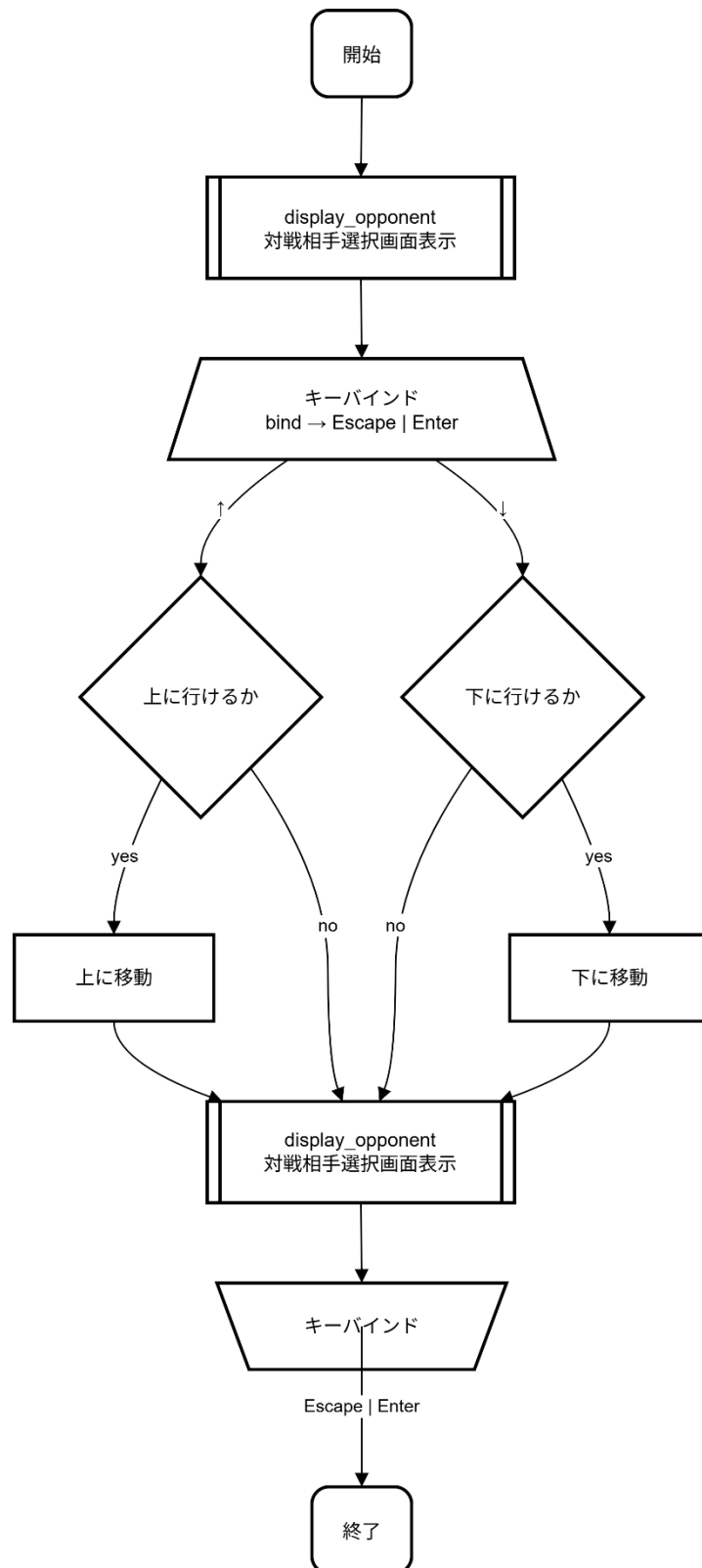
### 1.2.1 トップ画面モジュール



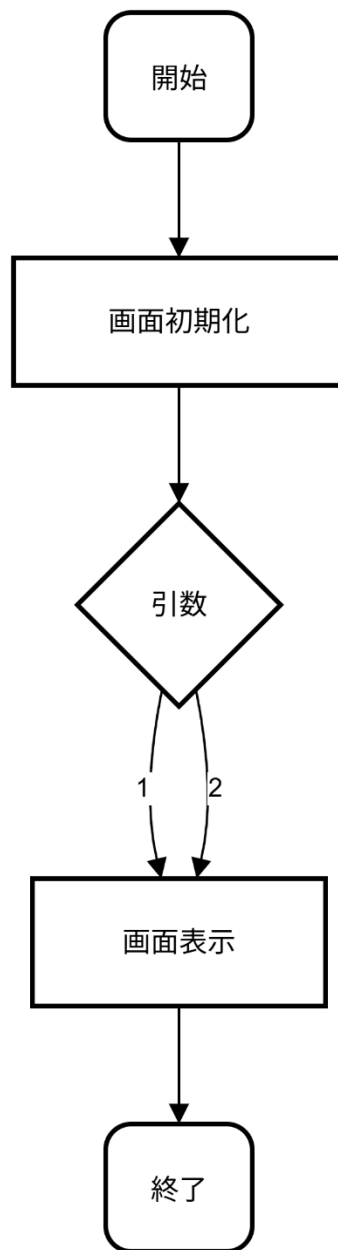
1.2.1.1 トップ画面表示関数



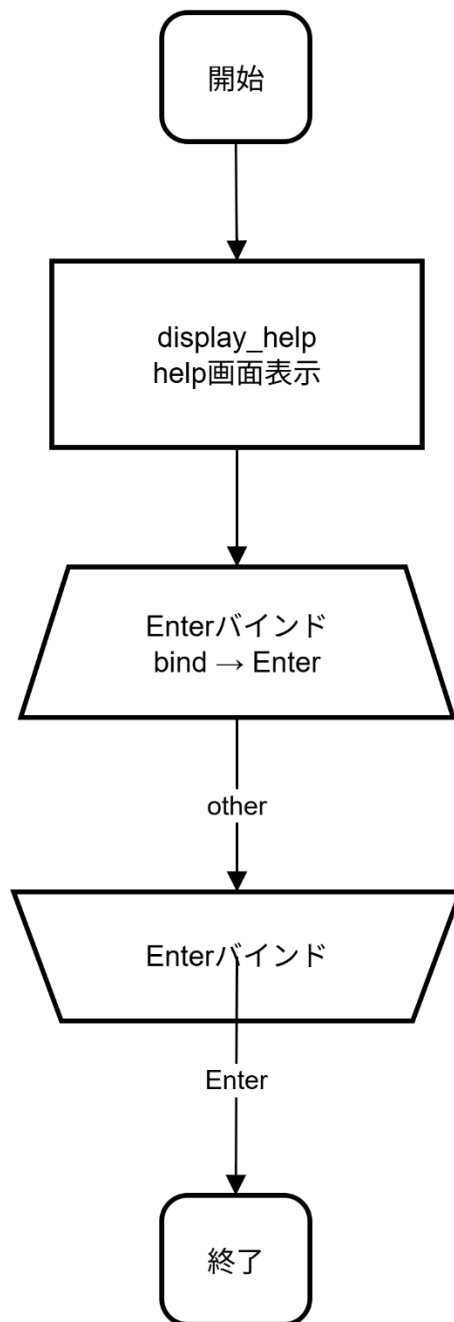
### 1.2.2 対戦相手選択画面モジュール



1. 2. 2. 1 対戦相手選択画面表示関数

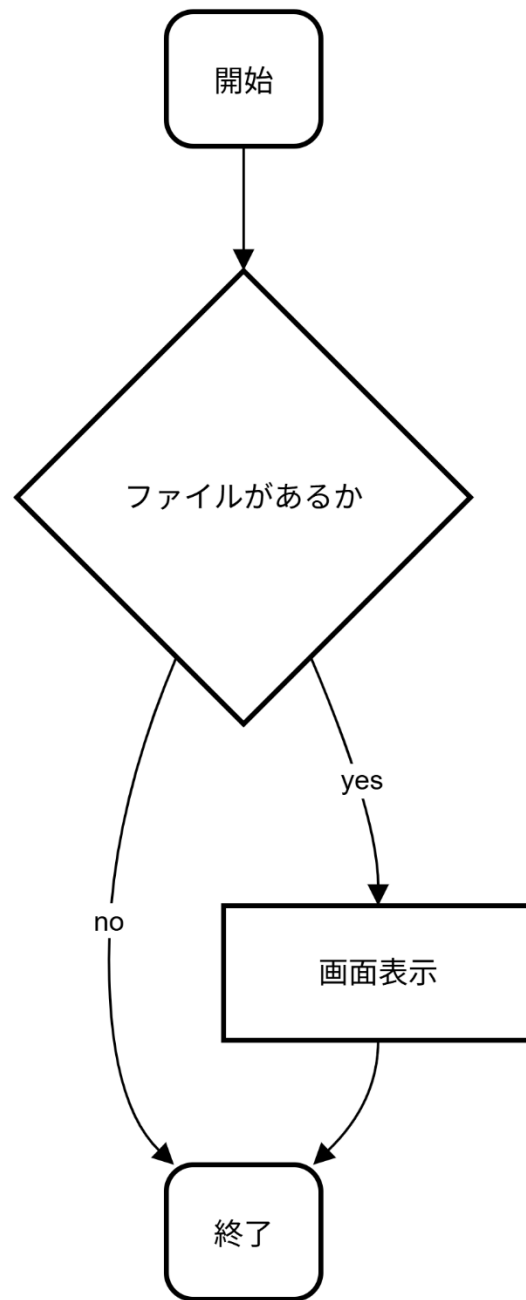


### 1.2.3 help 画面モジュール

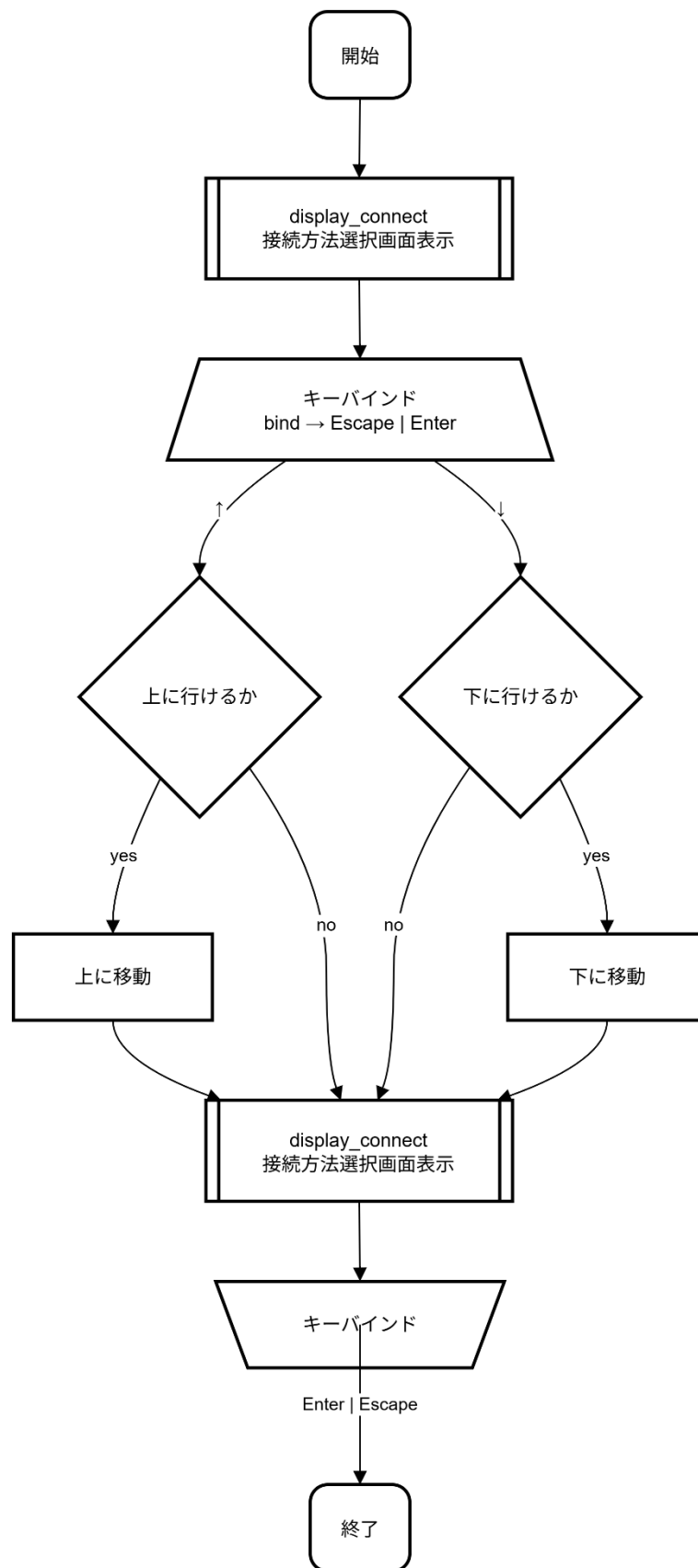


1.2.3.1

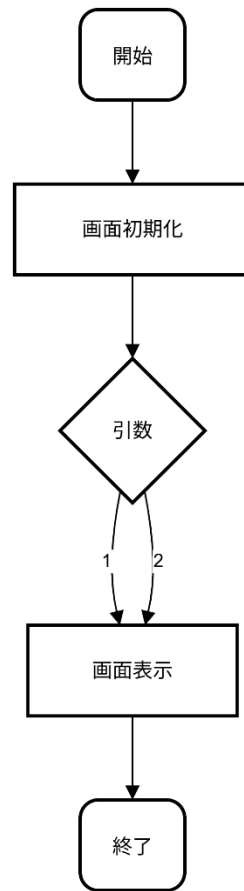
help 画面表示関数



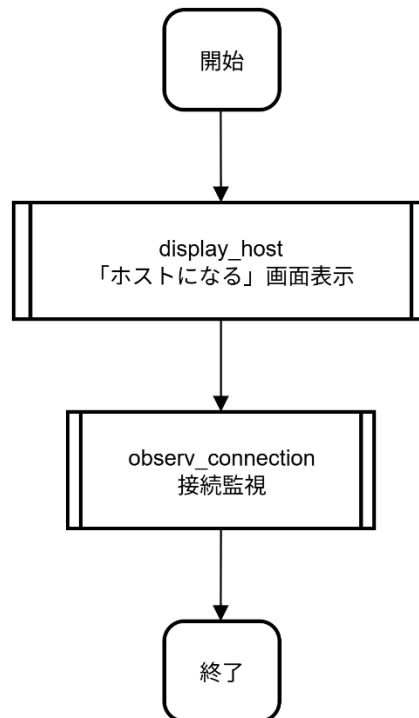
#### 1.2.4 接続方法選択画面モジュール



#### 1.2.4.1 接続方法選択画面表示モジュール

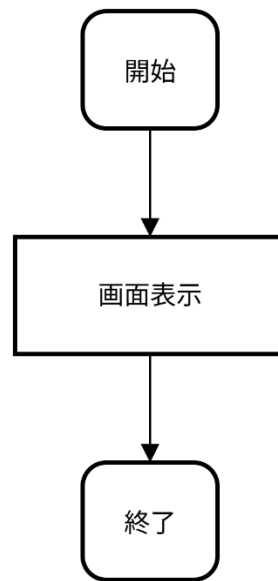


#### 1.2.5 「ホストになる」画面モジュール



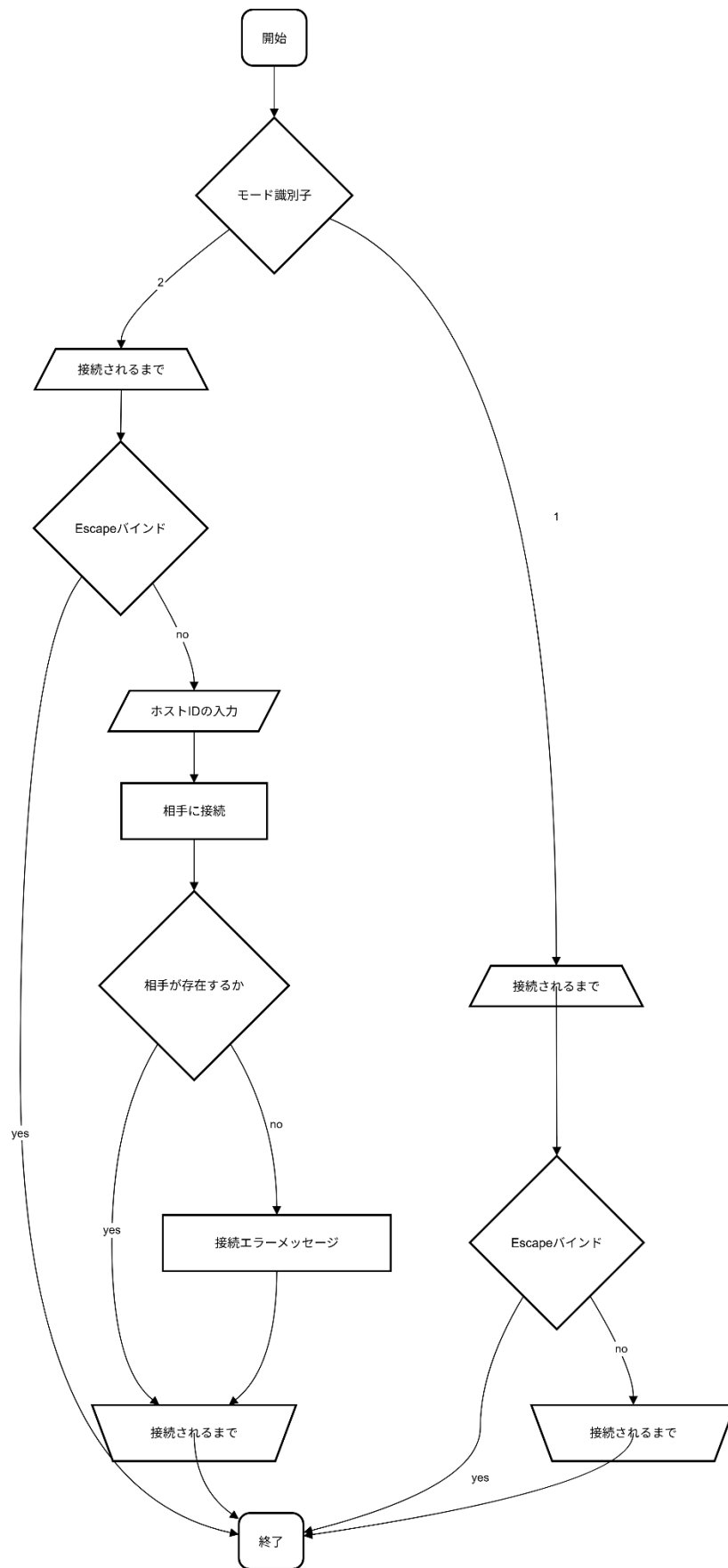


1.2.5.1 「ホストになる」画面表示関数

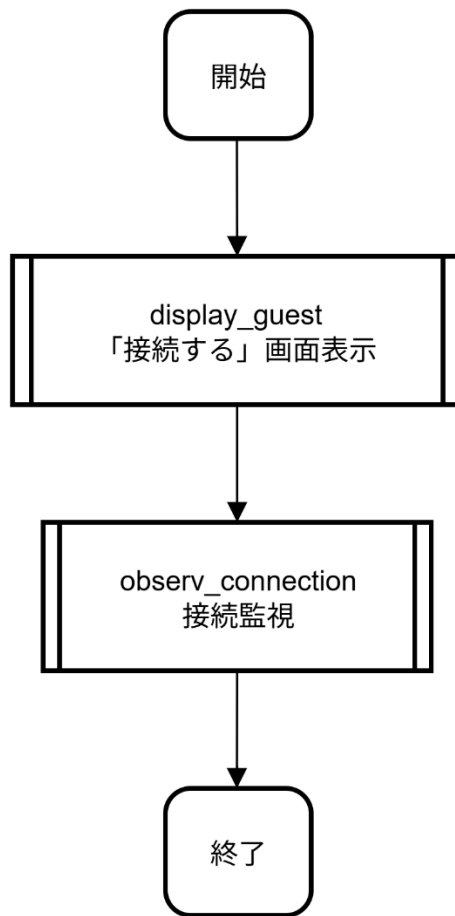


# 1. 2. 5. 2

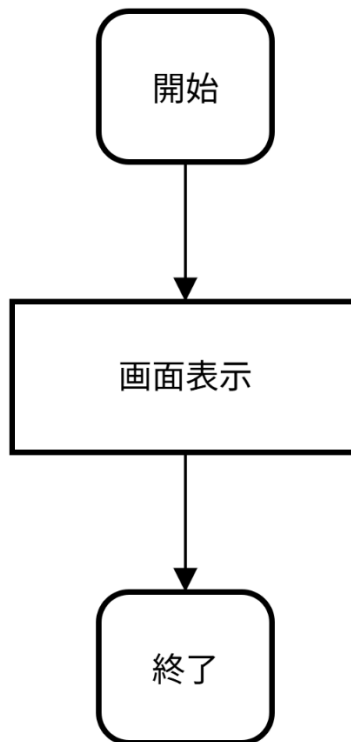
## 接続監視関数



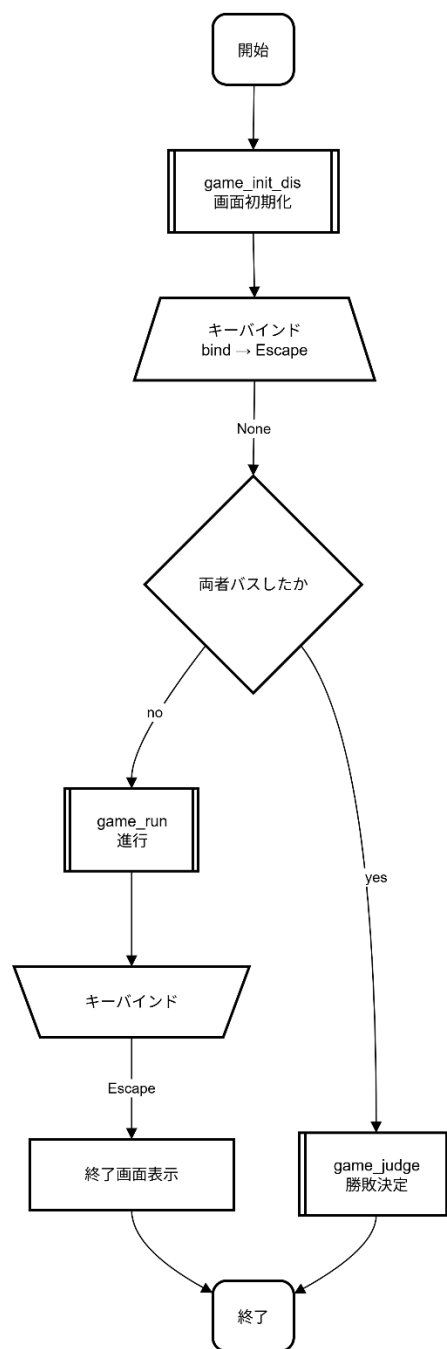
#### 1.2.6 「接続する」画面モジュール



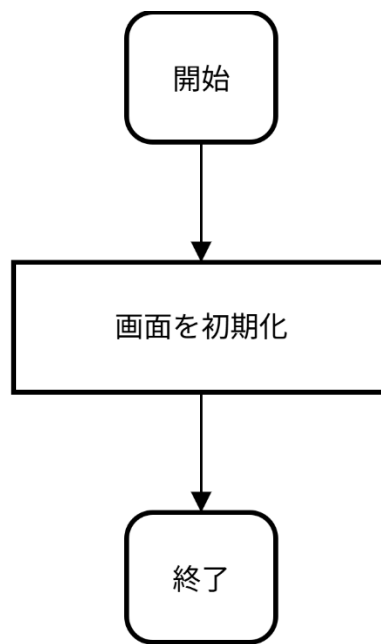
##### 1.2.6.1 「接続する」画面表示関数



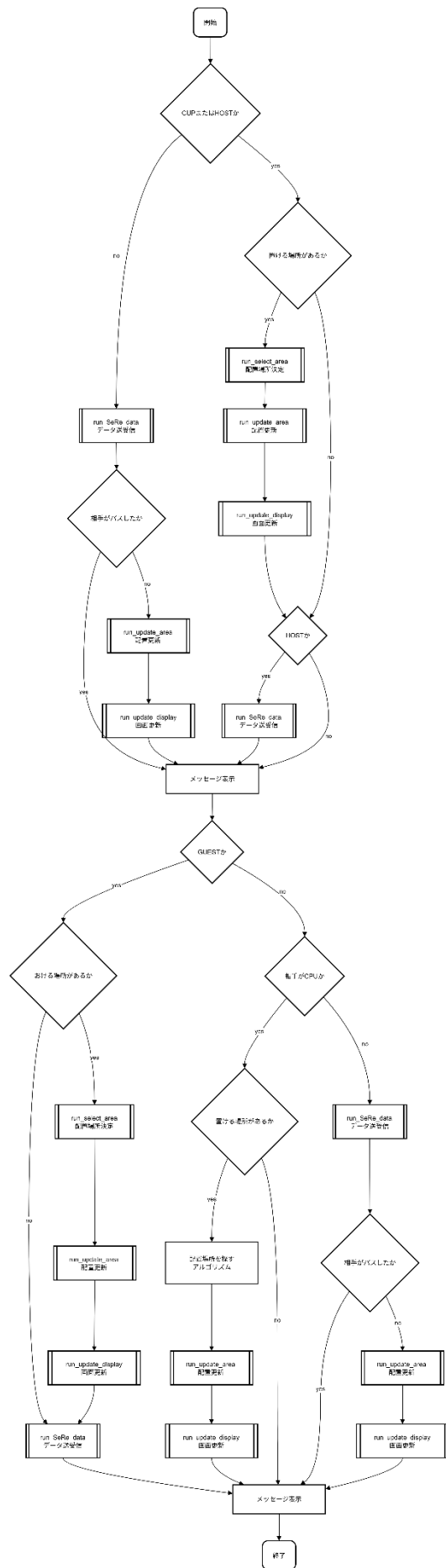
### 1.3 ゲーム開始モジュール



### 1.3.1 画面初期化

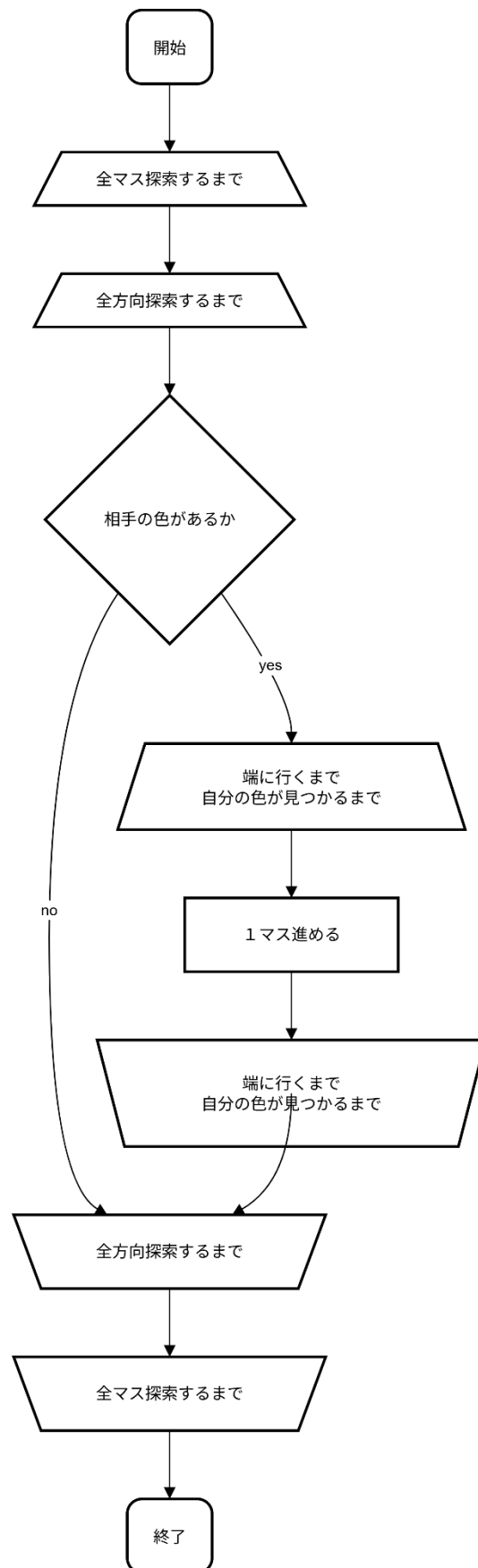


### 1.3.2 進行モジュール

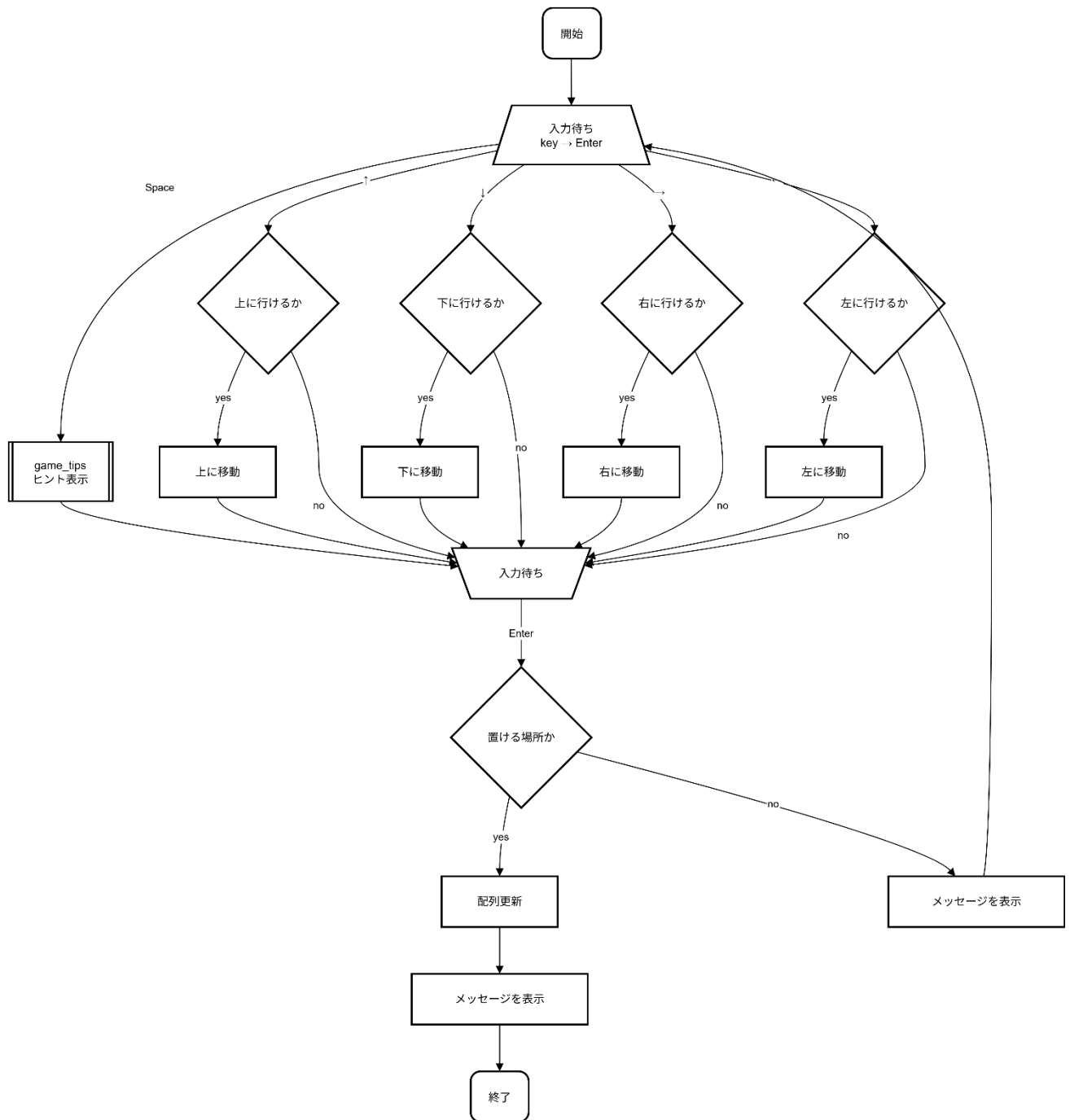


### 1.3.2.1

### 配置可能場所探索関数

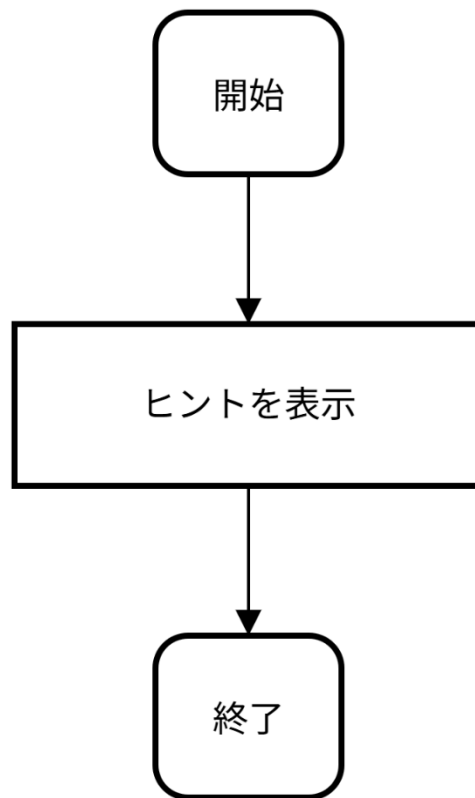


### 1.3.2.2 配置場所決定関数



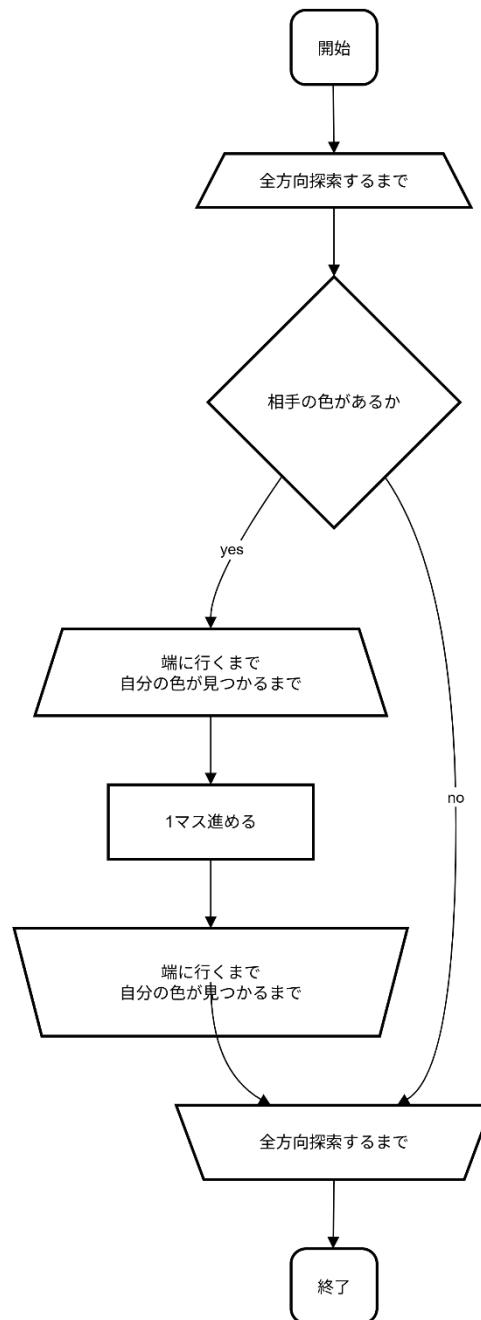


1.3.2.2.1 ヒント表示関数

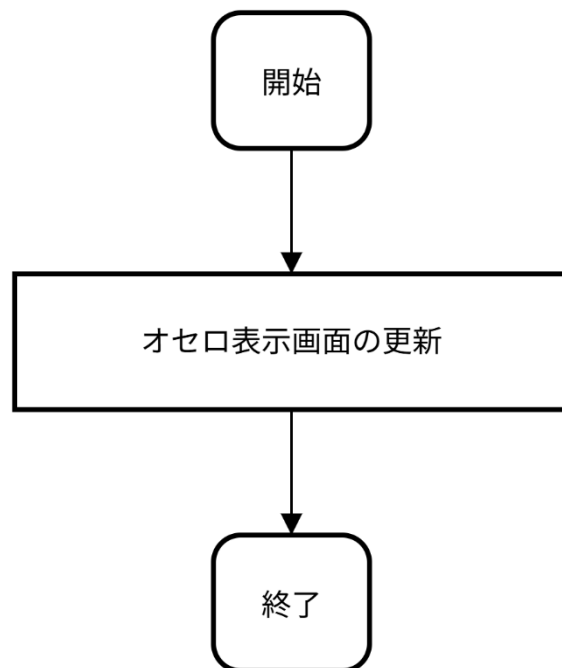


### 1.3.2.3

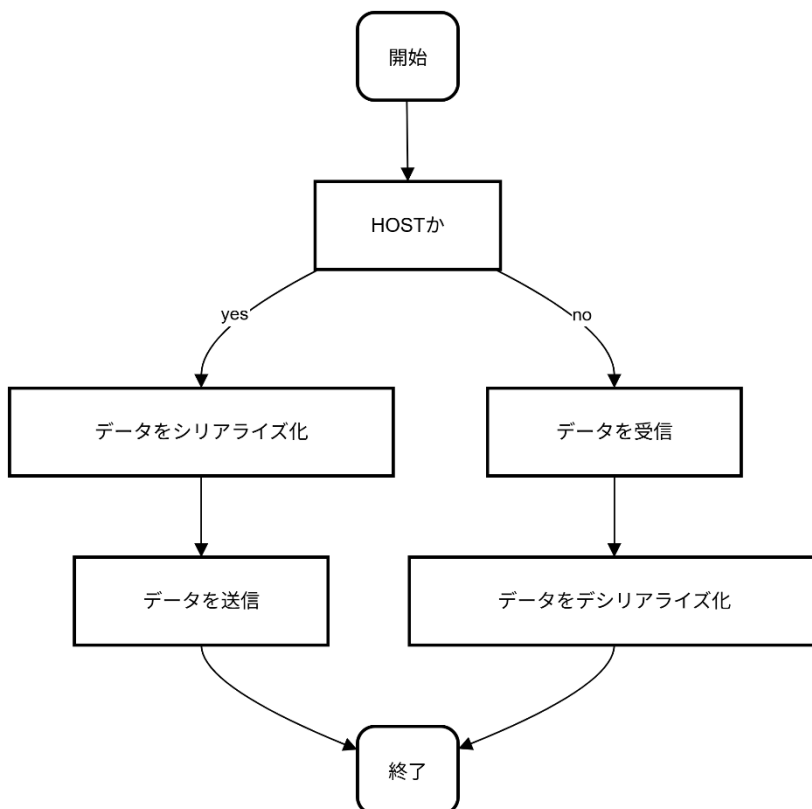
### 配置場所更新関数



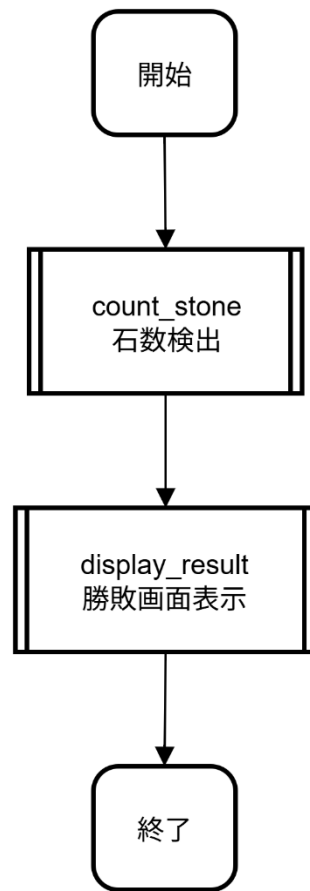
1.3.2.4 画面更新関数



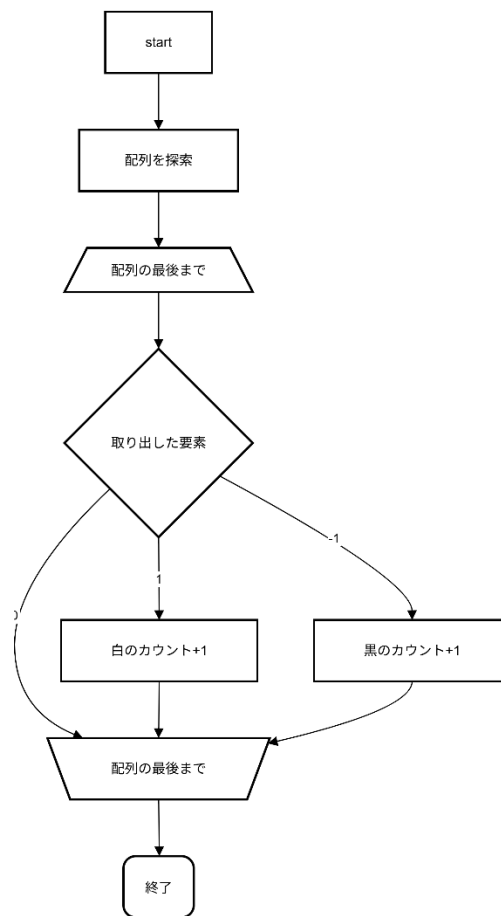
1.3.2.5 データ送受信関数



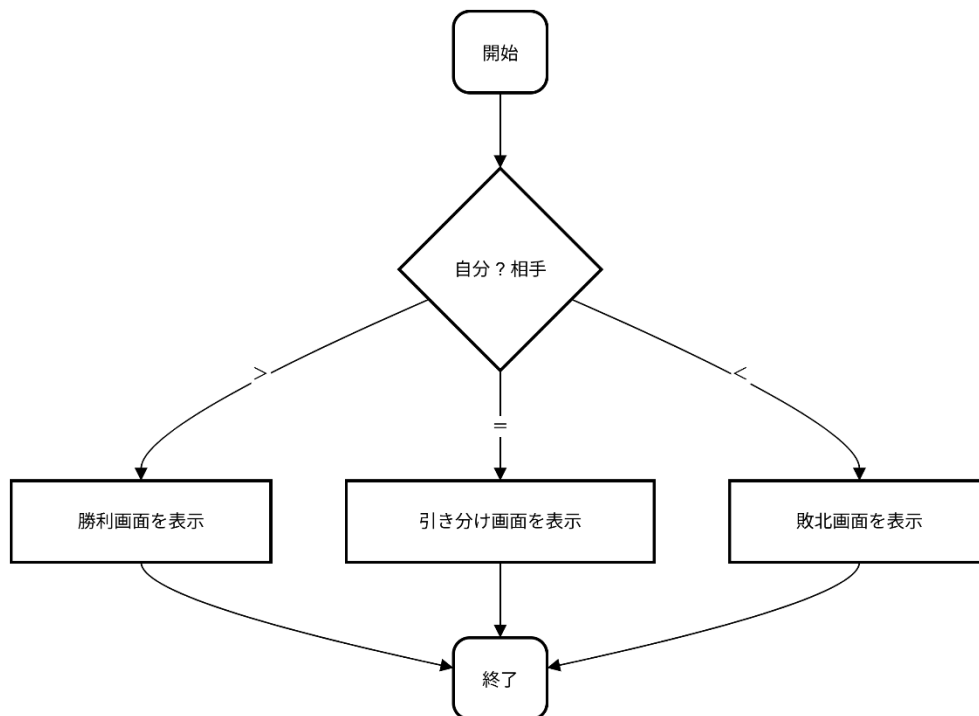
### 1.3.3 勝敗決定モジュール



#### 1.3.3.1 石数検出モジュール



#### 1.3.3.2 勝敗画面表示モジュール



## ■ 関数仕様一覧

このCプログラムにおいて、true を 1、false を 0 として typedef 宣言し boolean 型として扱うこととする。

### 1.1 メインモジュール

関数宣言	int main(void);		
引数	なし	戻り値	終了ステータス
概要	プログラムのエントリポイント		
処理内容	実行されると起動され、最初に画面サイズを設定し、メニュー選択ハンドラを起動する。 メニュー選択ハンドラが終了したらオセロを開始する。		

### 1.1 画面サイズ設定モジュール

関数宣言	void set_size(void);		
引数	なし	戻り値	なし
概要	画面のサイズを設定		
処理内容	画面サイズを判定して、必要なら画面サイズの強制変更をする。		

関数宣言	boolean size_resize(void);		
引数	なし	戻り値	既定サイズであるかどうか
概要	画面サイズを変更する関数		
処理内容	ユーザがウィンドウサイズを変更し、プログラムが判定をする		

### 1.2 メニュー選択ハンドラ

関数宣言	void handle_menu(void);		
引数	なし	戻り値	なし
概要	各メニューに遷移するハンドラ		
処理内容	ハンドラ内で無限ループして target に格納される変数で度の関数に渡すかを決定する。		

### 1.2.1 トップ画面モジュール

関数宣言	int screen_top(void);		
引数	なし	戻り値	target に格納する値
概要	トップ画面に関する操作をする関数		
処理内容	トップ画面を表示し、キーボード操作で選択する。		

関数宣言	void display_top(int position);		
引数	現在どこを選択しているかを示す変数	戻り値	なし
概要	トップ画面を表示する		
処理内容	引数に与えられた整数によってどこを選択状態にするかを決定し表示する		

### 1.2.2 対戦相手選択画面モジュール

関数宣言	int screen_opponent(void);		
引数	なし	戻り値	target に格納する値
概要	対戦相手選択画面に関する操作をする関数		
処理内容	対戦相手選択画面を表示し、キーボード操作で選択する。		

関数宣言	void display_opponent(int position);		
引数	現在どこを選択しているかを示す変数	戻り値	なし
概要	対戦相手選択画面を表示する		
処理内容	引数に与えられた整数によってどこを選択状態にするかを決定し表示する		

### 1.2.3 help 画面モジュール

関数宣言	int screen_help(void);		
引数	なし	戻り値	target に格納する値
概要	ヘルプ画面に関する操作をする関数		
処理内容	ヘルプ画面を表示し、キーボード操作で選択する。		

関数宣言	void display_help(void);		
引数	なし	戻り値	なし
概要	ヘルプ画面を表示する		
処理内容	ヘルプファイルがあるかを確認し、ヘルプ画面を表示する		

## 2.4 接続方法選択モジュール

関数宣言	int screen_connect(void);		
引数	なし	戻り値	target に格納する値
概要	接続方法選択画面に関する操作をする関数		
処理内容	接続方法選択画面を表示し、キーボード操作で選択する。		

関数宣言	void display_connect(int position);		
引数	現在どこを選択しているかを示す変数	戻り値	なし
概要	接続方法選択画面を表示する		
処理内容	引数に与えられた整数によってどこを選択状態にするかを決定し表示する		

## 1.2.5 「ホストになる」画面モジュール

関数宣言	int screen_host(void);		
引数	なし	戻り値	target に格納する値
概要	接続に関する操作をする関数		
処理内容	「ホストになる」画面を表示し、接続の監視をする。		

関数宣言	void display_host(void);		
引数	なし	戻り値	なし
概要	「ホストになる」画面を表示する関数		
処理内容	「ホストになる」画面を表示		



関数宣言	bool observe_connection(int mode);		
引数	モード識別子	戻り値	接続されたかを示す真偽値
概要	接続を監視する関数		
処理内容	接続を監視し、また、Esc キーバインドがあれば中断する。 モード識別子が「接続する」であればホスト ID の入力を促し接続する。		

#### 1.2.6 「接続する」画面モジュール

関数宣言	int screen_connection(void);		
引数	なし	戻り値	target に格納する値
概要	接続に関する操作をする関数		
処理内容	「接続する」画面を表示し、接続の監視をする。		

関数宣言	void display_connection(void);		
引数	なし	戻り値	なし
概要	「接続する」画面を表示する関数		
処理内容	「接続する」画面を表示する。		

#### 1.3 ゲーム開始モジュール

関数宣言	void start_game(void);		
引数	なし	戻り値	なし
概要	ゲーム本編に関する関数		
処理内容	画面の初期化をし、Space キーを常に監視し、ゲームの進行をする		

関数宣言	void game_init_dis(void);		
引き数	なし	戻り値	なし
概要	画面初期化する関数		
処理内容	初期状態の画面を構成し表示させる		

### 1.3.2 進行モジュール

関数宣言	int game_run(void);		
引数	なし	戻り値	パスをしたかのステータス 0:なし, 1:両者, -1:接続切れ
概要	ゲームの進行に関する関数		
処理内容	置ける場所を判定し配置、画面描写、相手の配置、描写をする。(CPU 判定は global 変数)		

関数宣言	void game_serch_placeable(int Go_mode);		
引数	どちらの色か表す変数	戻り値	なし
概要	配置可能場所を探す関数		
処理内容	全マス、全方向を相手の色があったら自分の色があるまで探索する		

関数宣言	void run_select_area(int coordinate[]);		
引数	値を返すための配列	戻り値	なし
概要	配置場所を選択する関数		
処理内容	矢印キーで移動し、配置する場所を決定しその座標を返す。		

関数宣言	void game_tips(void);		
引数	なし	戻り値	なし
概要	ヒントを表示する関数		
処理内容	ヒントを表示する		

関数宣言	void run_update_area(int coordinate, int player);		
引数	マス情報、プレイヤー情報	戻り値	なし
概要	マスの座標を更新する関数		
処理内容	ひっくり返すなどの処理をする		

関数宣言	void run_update_display(void);		
引数	なし	戻り値	なし
概要	マスの画面を更新する関数		
処理内容	画面をプレイヤー識別子に従って変更する。		

関数宣言	void run_SeRe_data(int coordinate[],int mode);		
引数	自分の座標+パスフラグ+モード識別	戻り値	なし
概要	相手と通信をして座標情報を送信、受信する関数		
処理内容	自分のパスフラグと自分の座標を送る、または受信する。 一連の処理が終わったら値を引数の配列に入れて返す。		

### 1.3.3 勝敗決定モジュール

関数宣言	void game_judge(void);		
引数	なし	戻り値	なし
概要	勝敗判定をする関数		
処理内容	石の数を検出し、勝敗の画面を表示する。		

関数宣言	int* count_stone(void);		
引数	なし	戻り値	相手と自分の石の数
概要	自分の相手の石の数を数える		
処理内容	配列の最後まで探索をし、配列の要素が 1:自分 , -1:相手 , 0:配置無 として数え上げ、その数を返す。		

関数宣言	void display_result(int* counts);		
引数	自分と相手の石の数	戻り値	なし
概要	勝敗画面を表示する		
処理内容	自分と相手の石の数を比較しそれに合う画面を出力する。		

## ■ 画面設計

画面の大きさは 136×30 とする。画面は自動で調整される。

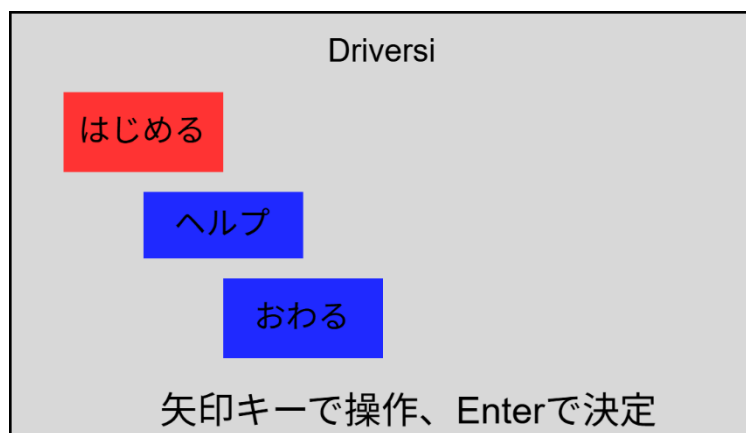


図 6 トップ画面

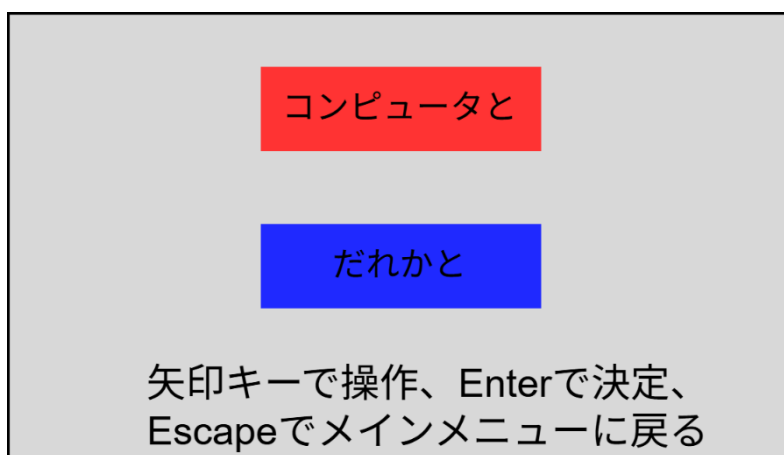


図 7 対戦相手選択画面

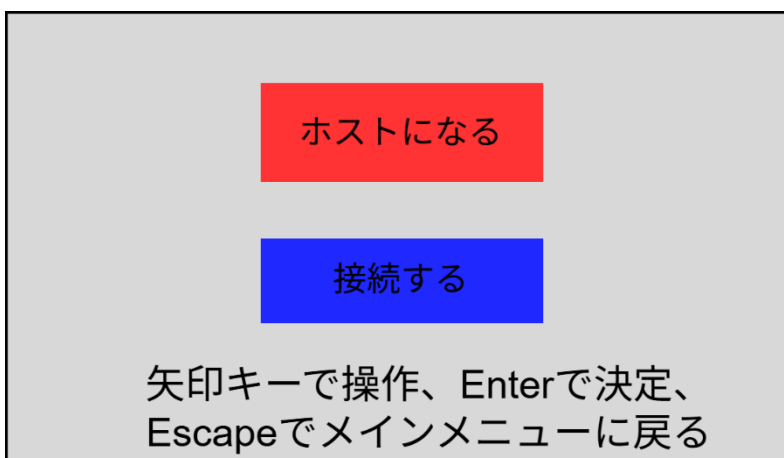


図 8 接続方法選択画面

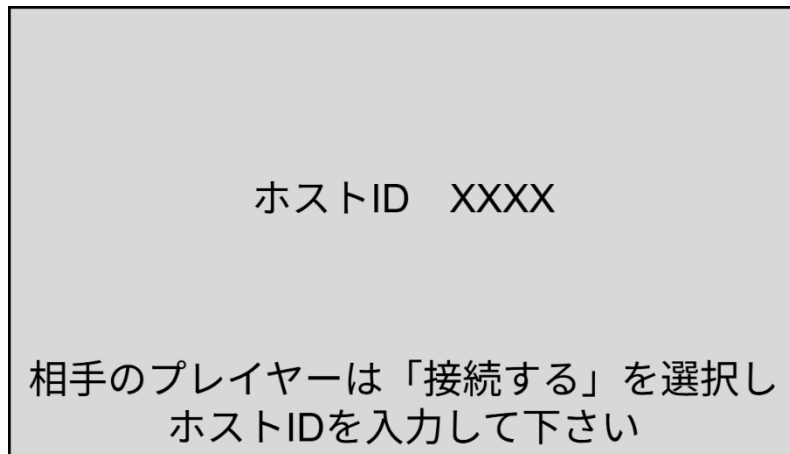


図 9 「ホストになる」の画面

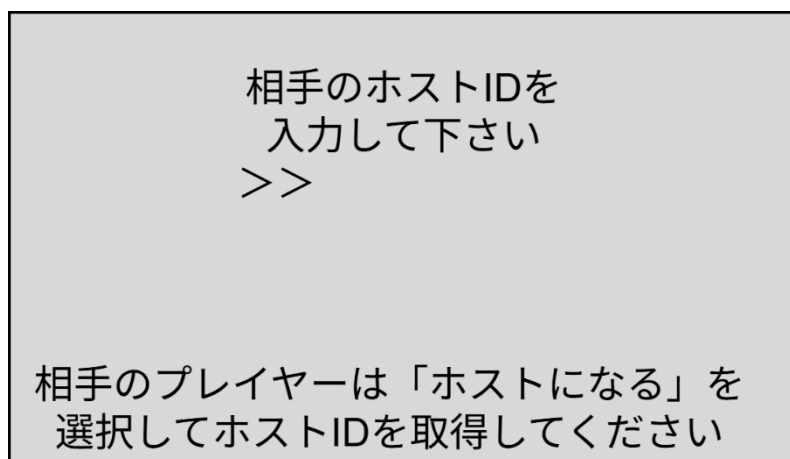


図 10 「接続する」の画面

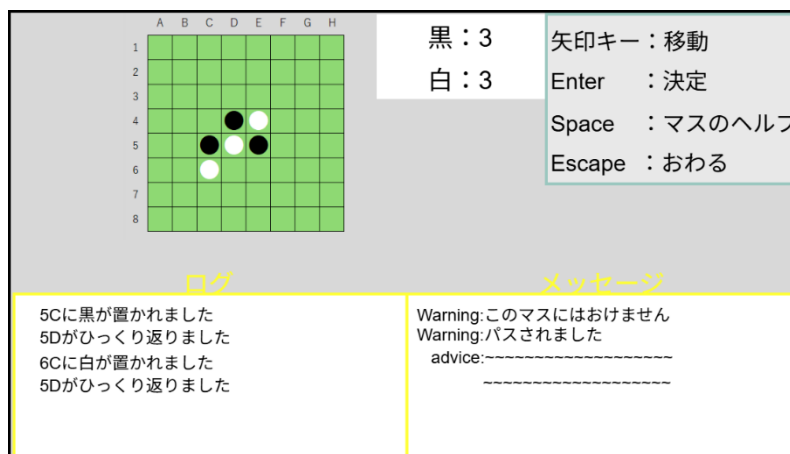


図 11 実際のゲーム画面

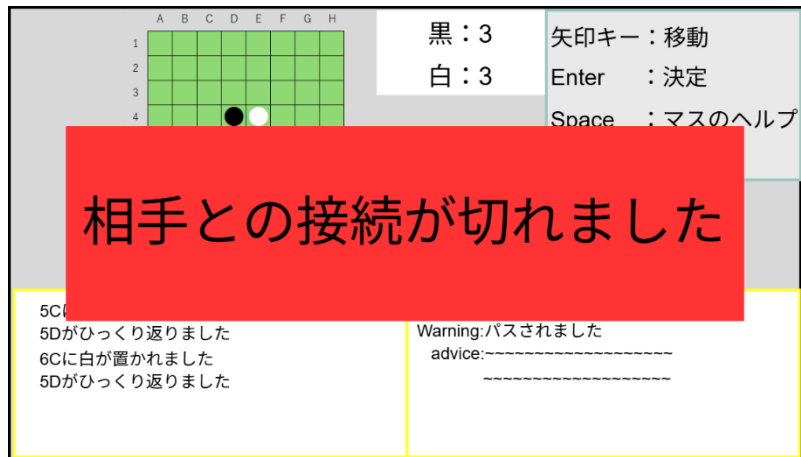


図 12 通信切断時の画面

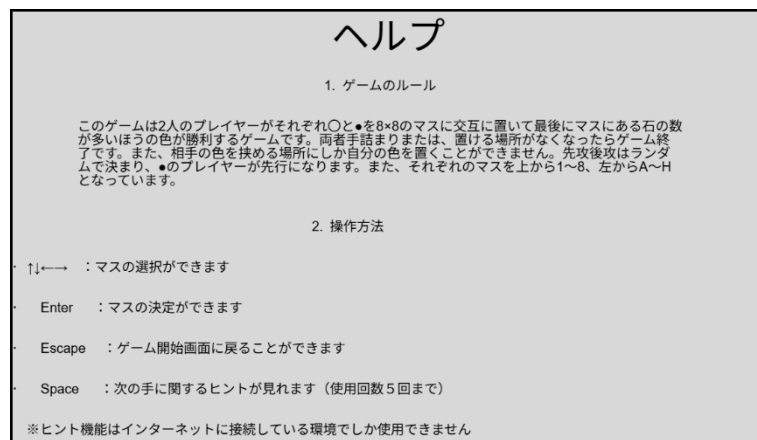


図 13 ヘルプ画面

## ■ 入出力ファイルの仕様

ファイルの文字コードは UTF-8 である。すべてのファイルを ./docs に配置する。

<入出力ファイル名>

- ・ヘルプファイル

help.txt

<入出力ファイルの仕様>

help.txt

ヘルプメッセージに関するデータが記述されている

## ■ エラーコード一覧

<エラーメッセージ一覧>

- 「./Dreversi」の後ろに引数などが入れられた場合、エラー出力で「オプションはありません」と出力される。
- ヘルプ項目選択時に、ヘルプファイルがない場合は「ヘルプを表示できません」と表示される。

- 「だれかと」モードで相手または自分の接続が切れた場合「相手との通信が切れました」と表示される。
- 「接続する」で相手が見つからなとき画面に「相手が見つかりません」と表示

## ■ 変更履歴

変更日	変更者	変更箇所	変更内容	変更理由
2025/06/13	Diceke	関数一覧	真偽値を int 型ではなく typedef 宣言で bool 型として扱う	型の認識がややこしくなるため
2025/06/13	Diceke	display_resullt 関数	内容を追加	内容が不完全
2025/06/13	Diceke	observe_connection 関数	接続先の入力追加	接続先が与えられていなかった
2025/06/13	Diceke	size_resize handle_menue 関数	追加	なかったため
2025/06/24	Diceke	ヒント機能	AI によるヒント機能を削除	セキュリティ的にもトークンを配布するファイル内に置くのはよくないため
2025/07/17	Diceke	モジュール構造図、フロー、関数定義	不足点を追加	不足していた点があったため