目 次

第１章　競技ルール

―概要―

この章では，プログラミングコンテスト競技部門のルールについて解説します．ゲームの基本となるため，ぜひ目を通しておいて下さい．また，中にはルールとして知ってるのと知らないのでは全然ゲームの難易度が違ってくるルールもあるので，隅から隅まで詳しく読んで少しでも優勝のために頑張って下さい．

1.1　説明に使う用語・記号について

ゲームに使う基本的な用語や記号と実際のゲームとの対応を説明します。

プレイヤ：ゲームに接続するプログラム (つまり，今回の大会で皆さんに書いてもらうプログラム) とそれを使って対戦する人です．

サーバ：接続されて，対戦の状況などを管理するプログラムのことです．

COOL：ゲームの先攻側です．
サーバに先に接続したプログラムが先攻になります．

HOT：ゲームの後攻側です．
サーバに後から接続したプログラムが後攻になります．

ターン：自分が１回行動する一連の動作 (getReady，周辺情報の取得，メソッド，周辺情報の取得) をターンと呼びます．

フィールド：ゲームを行う場所のことです．

周辺情報：自分の居るマスを含めた周囲 9 マスの状態とゲームが終了しているか否かの情報のことです，

メソッド　自分のターン中に１つ行える動作のことです．

また，この章ではメソッドの説明など，言葉だけではイメージが掴みづらい部分が多いです．なので，これから使われる図と用語について対応表を書いておきます (そして，COOL を自分として書いていますが，HOT でも同様です)．

自分 (**COOL**) 

相手 (**HOT**) 

普通の床 

ブロック 

アイテム 

1.2　画面のサンプル

具体的なサンプルをもとに実際にどのような配置になるかを見てみたいと思います．実際の競技画面は以下の様になります．

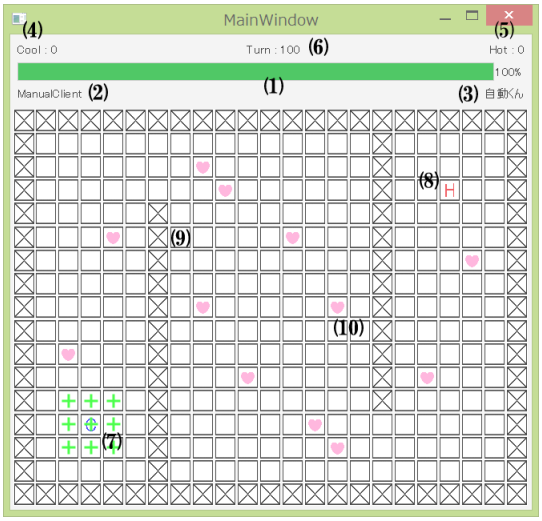

図 1.1: 実際の画面

(1) 制限ターン数バー

(2) COOL の名前

(3) HOT の名前

(4) COOL が獲得したアイテムの個数

(5) HOT が獲得したアイテムの個数

(6) 制限ターン数 (ゲーム開始からは終了までのターンのカウントダウンとなる)

(7) COOL のアイコン

(8) HOT のアイコン

(9) ブロック

(10) アイテム

1.3 ルール

1.3.1 基本的なルール

- ⋆ 競技は，サーバに接続された 2 人のプレイヤによる 1 対 1 での対戦になります．
- ⋆ 先に接続した側を先攻になり，後に接続した側が後攻になります．
- ⋆ 勝利条件 (2.3.2 節を参照) を先に満たしたプレイヤが勝利となります．
- ⋆ 競技に使用するフィールドは，15 × 17(縦 15 マス，横 17 マス) です．
- ⋆ アイテム・ブロックは，マップの中心から点対称になる様に配置されます．
- ⋆ アイテムを取得したとき，元居たマスは以下の様になります (下記の図を参照)．

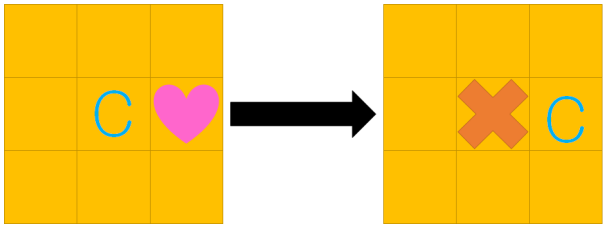

図 1.2: 動作の例

1.3.2 勝利条件

- ⋆ 決められた，制限ターンになったときに，
  アイテムを相手よりも多く集めている．
- ⋆ ブロックを対戦相手の上に乗せる．
- ⋆ 対戦相手の上下左右をブロックで囲み，動けなくする．
- ⋆ 対戦相手がブロックの上に移動する．
- ⋆ 対戦相手が試合続行不可能になる．

この条件で勝負がつかなかった場合は引き分けとなります．

1.4　ゲームで行える行動

1.4.1　getReady

自分のターンがくるまで待機します．自分のターンが来たとき，サーバから与えられる周辺情報を取得します．

また，ターンの間で起動した場合，試合が続行出来ないのでターンの間は起動しない様にして下さい！

1.4.2　メソッド

メソッドは，“[動作][Up, Right, Left, Down のどれか]” という形で書かれています (例 :“walkRight”)．今回，載せる図などはすべて “Right” での動作です．ほかの “Up” などは適時読み替えて下さい．

walk

指定された上下左右のどれかの方向に移動します．メソッド実行後に取得される周辺情報は移動後の周辺情報です．

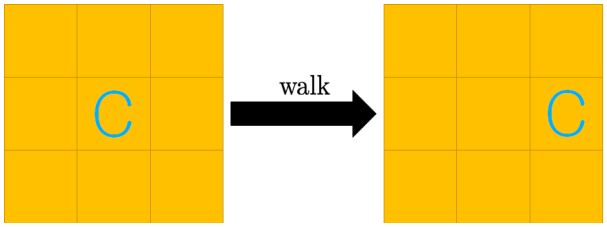

図 1.3: walk のイメージ (walkRight の場合)

look

指定された上下左右のどれかの方向に対して，正方形状に 9 マスの情報を取得します (下記の図を参照)．メソッド実行後に取得される周辺情報は look で取得した情報です．周辺情報割当ての順番は図に割り振られている番号と一緒です．

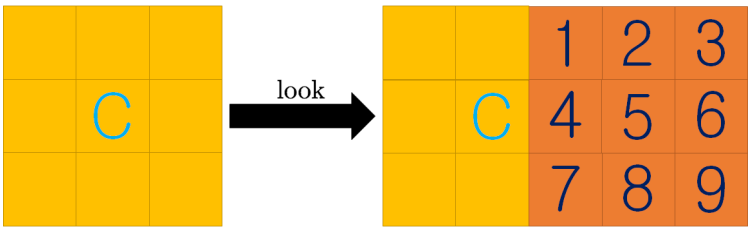

図 1.4: look のイメージ (lookRight の場合)

search

指定された上下左右のどれかの方向に対して，直線上に 9 マスの情報 (下記の図を参照) を取得します．メソッド実行後に取得される周辺情報は search で取得した情報です． 周辺情報割当ての順番は図に割り振られている番号と一緒です．

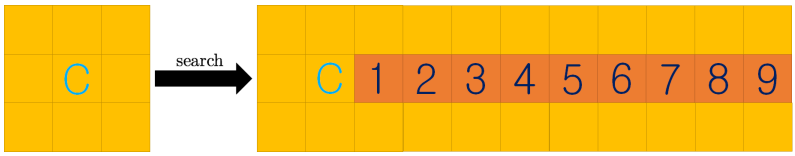

図 1.5: search のイメージ (searchRight の場合)

put

指定された上下左右のどれかの方向に対して，ブロックを配置します．メソッド実行後に取得される周辺情報は，ブロックをおいた後の周辺情報です．

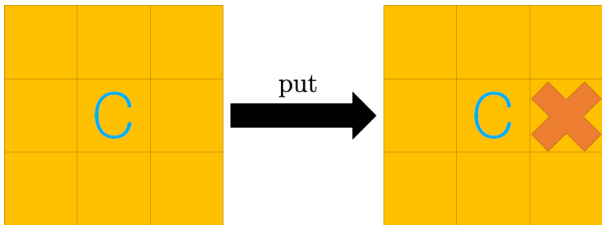

図 1.6: put のイメージ (putRight の場合)

1.5　周辺情報

サーバから送信される周辺情報については，具体的には以下の通りの様な仕様となっています．

(1) 10 桁の 0 から 3 までの数字です．

(2) 1 桁目には，ゲームの状態 (終了しているかどうか) を表しています．0 が終了状態で，1 が続行している状態です．

(3) 2 桁目から 10 桁目には対応する自分の周囲 9 マスの情報 (図 2.7 参照) が保存されています (5 番目が自分の現在居る位置です)．また，各桁に 0 から 3 までの数字が割り当てられて，以下に示す様な意味になります．

図 1.7: 周辺情報の割当図

表 1.1: 周辺の状態割り当て表

割り当てられている数値	状態
0	なにもなし
1	相手プレイヤがいる
2	ブロックがある
3	アイテムがある

以下に，実際の状態と値の対応表を書きます．

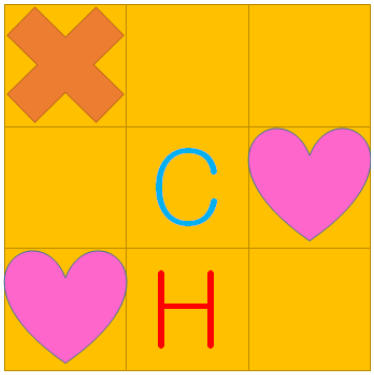

図 1.8: 周辺情報の例

表 1.2: 周辺情報の値

周辺情報の桁	値
1	1
2	2
3	0
4	0
5	0
6	0
7	3
8	3
9	1
10	0

1.6 具体的な通信の仕様

Warning!：この章は，自分で新しくライブラリを作成する人向けの章です．それ以外の人は読まなくても問題有りません．

別の言語などで，独自のライブラリを構築する場合，以下の仕様にのっとって作ればサーバとの通信が行えます．

(1) サーバとの接続を確立 (接続のポート番号は前述の値を参照して下さい)，

(2) サーバへチーム名を送信します (ここで，先頭 5 文字目以降は無視されます)．

(3) サーバが送信する “@” を受け取ります．

(4) “gr\r\n” をサーバに送信します (これを，本テキストでは getReady と呼んでいます)．

(5) サーバ側から送信される 10byte の文字列を受け取ります (これを，本テキストでは周辺情報と呼び，実装では value に格納しています)．例としては，“1000000000” となります．

(6) サーバにメソッドを表現する 2 文字とその末尾に “\r\n” を付加した文字列を送信します．表現の対応表は，

表 1.3: メソッドの表現対応表

	Up	Right	Left	Down
walk	“wu”	“wr”	“wl”	“wd”
look	“lu”	“lr”	“ll”	“ld”
search	“su”	“sr”	“sl”	“sd”
put	“pu”	“pr”	“pl”	“pd”

(7) (5) と同様に周辺情報を受け取ります．

(8) サーバに “#\r\n” を送信します．

(9) 以降，周辺情報の最初の文字が “0” になるまで，(4) からここまでを繰り返す．

以下に概要図を示すので，これを利用して作成に役立てて下さい．

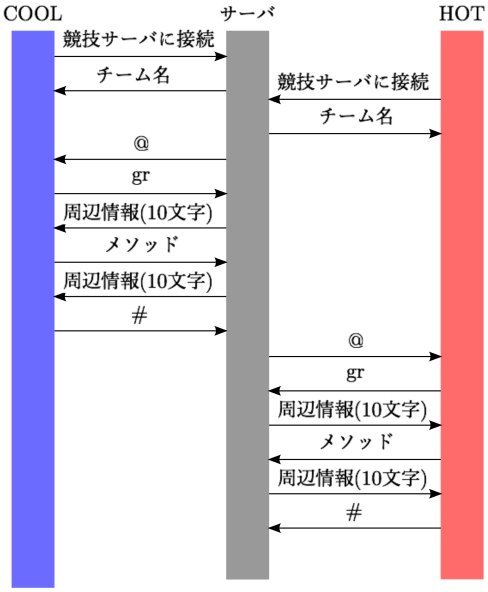
COOL
サーバ
HOT
競技サーバに接続
チーム名
競技サーバに接続
チーム名
@
gr
周辺情報(10文字)
メソッド
周辺情報(10文字)
#
@
gr
周辺情報(10文字)
メソッド
周辺情報(10文字)
#

図 1.9: 動作模式図

第2章　プログラムの使用方法

では，プログラムを書いていく方法を学びます．

2.1　プログラム作成から実行の流れ

プログラミングは基本的に一連の流れによって，作成されます．

(1) プログラムソースを書く．

(2) コンパイルする．

(3) コンパイルが成功したら次へ行く．もし，失敗したら (1) へと戻ってやり直す．

(4) プログラムを実行する．今回の大会は以下の通りに実行する．

  i サーバプログラムを起動する．

  ii COOL(先攻側のプログラム) を起動する．

  iii 以下のいずれかで対戦相手を起動する．

    (⋆) HOT（後攻側のプログラム）を起動する．

    (⋆) 自動くんを選択する．

    (⋆) ManualClient を選択する．

普通ならばこれらを黒い画面に文字を打ち込んで行います．しかし今回の大会では，ダイアログのボタンをクリックしていくだけで実行できます．
それでは，どのように起動させればいいか説明します．

(1) プログラムをコンパイルします．[F5] キーに「コンパイル+実行」が割り振られているので，それを使用します．[dialog] という名前の画面が出てきたら問題なく成功です．

  逆に，[結果レポート] という画面が出てきた場合は，プログラムのどこかが間違っています．[結果レポート] 画面を閉じて，プログラムを見直してみましょう．

  問題なく [dialog] 画面が出てきたら次へ進みます．

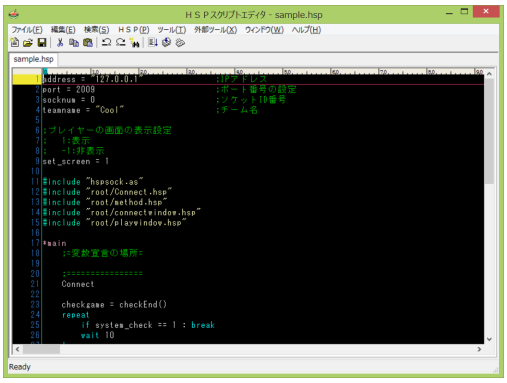

図 2.1: HSP スクリプトエディタの画面

図 2.2: サーバ接続画面

(2) サーバーを起動します．まずは HOT，COOL を接続させます．どちらも手順は同様ですので，今回は COOL の場合で説明します．

サーバ接続画面の説明です．

(1) COOL の接続状態
(2) HOT の接続状態
(3) COOL の名前
(4) HOT の名前
(5) COOL の IP 番号
(6) HOT の IP 番号
(7) COOL のポート番号
(8) HOT のポート番号
(9) COOL の選択ユーザー
(10) HOT の選択ユーザー
(11) COOL の接続設定ボタン
(12) HOT の接続設定ボタン
(13) マップファイル選択ボタン
(14) マップ作成ボタン
(15) ゲーム開始ボタン

i ユーザーの種類を選びます．COOL の中にある [TCP ユーザー] をクリックし，出てくる 3 つの項目から 1 つ選択します．それぞれの項目は次のようになっています．

TCP ユーザー 私たちのプログラムから接続する場合に選択します．

自動くん 既にプログラムが組まれている，常に一定のメソッドを行うプレイヤーです．自分のプログラムを確認するときにとりあえずの対戦相手として選択すると良いです．

ManualClient ユーザーがその場でメソッドを入力する場合に選択します．対戦相手の動きを自分で操作したい場合に選択すると良いです．

[自動くん]，[ManualClient] を選択した場合は，接続完了です．

ii [TCP ユーザー] を選択した場合は，ポート番号を確認し，[待機開始] をクリックします．これで，サーバが COOL の接続を待っている状態になります．
もし何らかの理由で接続待ちを解除したい場合には，[待機終了] をクリックしてください．再度 [接続開始] を押すことで接続待ちとなります．

iii クライアント側の準備を行います．

図 2.3: クライアント接続画面

クライアント接続画面の説明です．

(1) クライアントの接続状況　(4) 接続開始ボタン
(2) 接続 IP
(3) 接続するポート番号　(5) ゲーム開始ボタン

まず，プログラムの上の方にある **port という変数に，2009 という数字が入れられていることを確認します．**(HOT の場合は port=2010 となります．)
次に，プレイヤーの画面設定を行います．ゲームを開始すると，私たちが得た情報で構成された画面が現れます．もしもその画面が必要のない場合には set_screen という変数を-1 と書き換えます．表

示する場合は 1 と書き換えます.
それぞれの数字が正しいことを確認したら，[F5] キーを押してプログラムを実行します．[dialog] 画面が出てきます.

iv サーバ側が接続待機状態になっていることを確認して，クライアント側の [dialog] 画面の [接続開始] をクリックします．接続に成功すると「接続しました」というダイアログが表示されるので，それを閉じます.

v サーバ側の画面を見ると，COOL の「状態」が「準備完了」となっていれば COOL の準備は終わりです.

同様の手順で HOT の接続を行います.

(3) マップを用意します.

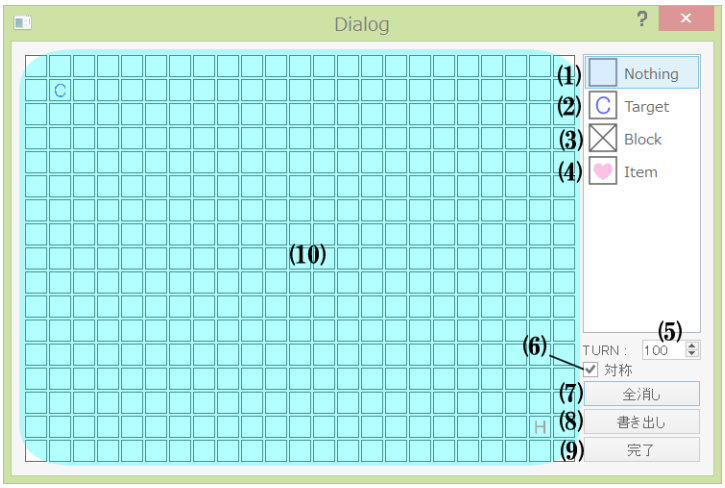

図 2.4: サーバ接続画面

マップ作成画面の説明です.

(1) 床アイコン
(2) プレイヤーアイコン
(3) ブロックアイコン
(4) アイテムアイコン
(5) ターン数設定
(6) 対称チェック
(7) 全消しボタン
(8) 書き出しボタン
(9) マップ作成完了ボタン
(10) 作成マップ

サーバ側の画面にある，[MapEdit] をクリックします．マップ作成画面が現れます．右側にあるアイコンを選んで，マップの好きな位置をクリックすると，選択したアイコンがマップ上にプロットされます.

「対称」にチェックマークがついていると，アイコンをプロットしたときにマップの中心に対して対称な場所にもアイコンがプロットされます．チェックマークを外すと，クリックした位置にのみアイコンがプロットされます．

サーバを起動するたびにマップを書くことは大変ですし，プログラムの動作の違いを見るには適していません．そこで，書いたマップを保存する方法，マップを呼び出す方法を確認します．

マップの保存

(i) マップの保存方法です．マップが書き終わったら，マップ作成画面右側にある [書き出し] ボタンをクリックします．わかりやすい場所に，名前を付けたマップデータを保存します．

(ii) マップの呼び出し方です．サーバダイアログの [SERVER] という枠の中を見てください．「MapData...」と書かれている枠の右側にあるボタンをクリックします．そこから保存したマップを選択し，開きます．

(4) ゲームを開始します．サーバ側の画面で [ゲーム開始] ボタンをクリックします．サーバの準備が出来ました．クライアント側の画面で [ゲーム開始] ボタンをクリックします．ゲームが開始します．

(5) ゲームの終了します．クライアントの画面を閉じてからサーバの画面を閉じてください．

これで一連の流れの確認ができました．

2.2　質問する前に

思ってもいない画面が表示された場合，次に当てはまるかどうか確認してください．

Q.1 **サーバを起動したら管理者権限云々という画面が出てくる**
気にせず閉じてください．

Q.2 **ゲームが終わって画面を閉じようとしたら，「ターンの開始を受信できませんでした」というメッセージの後に [Error] という画面が出てきた**
問題ないです，気にせず閉じてください．

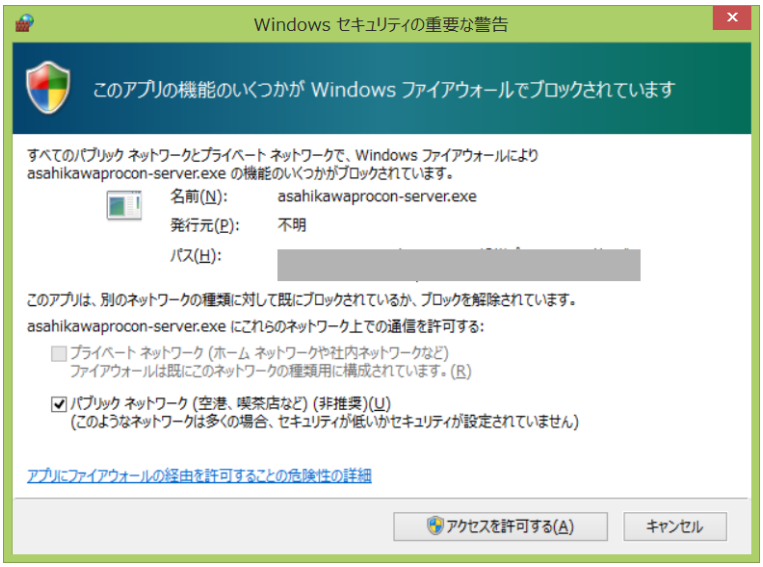
Windows セキュリティの重要な警告
このアプリの機能のいくつかが Windows ファイアウォールでブロックされています
すべてのパブリック ネットワークとプライベート ネットワークで、Windows ファイアウォールにより asahikawaprocon-server.exe の機能のいくつかがブロックされています。
名前(N): asahikawaprocon-server.exe
発行元(P): 不明
パス(H):
このアプリは、別のネットワークの種類に対して既にブロックされているか、ブロックを解除されています。
asahikawaprocon-server.exe にこれらのネットワーク上での通信を許可する:
プライベート ネットワーク (ホーム ネットワークや社内ネットワークなど)
ファイアウォールは既にこのネットワークの種類用に構成されています。(R)
パブリック ネットワーク (空港、喫茶店など) (非推奨)(U)
(このようなネットワークは多くの場合、セキュリティが低いかセキュリティが設定されていません)
アプリにファイアウォールの経由を許可することの危険性の詳細
アクセスを許可する(A)
キャンセル

図 2.5: 警告画面