

条件の変更にロバストなデジタルカーリングの改良

森健太郎^{†1} 伊藤毅志^{†1}

概要:「デジタルカーリング」は、カーリングのゲームをコンピュータ上で再現したシステムである。このシステムがカーリング AI 研究の発展に大きく寄与したが、実環境の状況の要素やルールの変更への対応が困難であることなどの問題点も指摘されていた。本研究では、これらの問題に対処するためにシステム全体を再構築し、ルール変更に対応できるシステムの構築を行った。また、プレイヤー毎の技量差を導入することで、より現実に近いカーリングを表現できるようにした。

キーワード: カーリング, デジタルカーリング, シミュレータ, 試合進行クラス

Improvements to Digital Curling System

KENTARO MORI^{†1} TAKESHI ITO^{†1}

Abstract: "Digital curling" is a system that plays Curling's games on a computer. Although this system greatly contributed to the development of curling AI research, problems such as difficulty in dealing with changes in elements and conditions of the real environment situation were also pointed out. In this research, in order to solve these problems, we reconstructed the whole system and realized a system which can respond flexibly to rule change. In addition, we realized more realistic curling by expressing skill difference for each player.

Keywords: curling, Digital Curling, simulator, game process class

1. はじめに

「デジタルカーリング」とは、カーリングの戦略のみを議論することを目的として、北清らが 2014 年に開発し、発表したカーリングプラットフォームである[1][2]。このシステムは 2D 物理シミュレータを用いて氷上のストーンの動きを単純な物理法則で模倣することで、カーリングゲームのシミュレーションを行う。このシステムを介して人間のプレイヤーだけでなく、カーリングショットを生成するプログラム（以下、カーリング AI と呼ぶ）とのオンライン上での対戦も可能となり、年に数度大会も開催されている[3]。このシステムが広く公開されたことで、多くのカーリング AI が同じ条件で対戦されるようになり、カーリング AI の発展に大いに進展した[4][5]。

このシステムは上述のような成果をもたらした一方で、大会参加者やカーリングプレイヤーから「氷の状態の変化などの実環境のカーリングの要素を表現しにくいこと」や「ルールの変更への対応が難しいこと」などの問題点も指摘されている。

本研究では、これら従来のシステムの問題点に対する改良を行い、ルールの変更に対応できるだけでなく、より詳細な条件（氷の状態の変化やプレイヤーの技量など）の変化にも対応が可能な新たなシステムの構築を行った。

2. デジタルカーリング

2.1 概要

カーリングの通常ルールに基づいてローカルコンピュータ上にカーリング AI（または人間）同士の試合を行えるゲームサーバが、「Digital Curling - Client」として公開されている[10]。ゲームサーバには物理シミュレータが Dynamic Link Library (DLL) 形式で同封されている。ユーザは GUI から試合条件とカーリング AI のパスを指定することで、カーリング AI 同士の対戦を行うことが出来る。

図 1 にこれまでのシステム（以下、従来システムとする）の概要図を示す。従来システムはゲームサーバと物理シミュレータから構成される。このほかにも試合ログの再生機能なども含むが、本論文では試合の実行に関する部分についてのみ扱うため、ここでは割愛する。

ゲームサーバは、グローバル領域に定義された試合進行を行う関数群から構成され、同様にグローバル変数として定義された各種パラメータ群を参照しながら、試合進行の制御を行う。ゲームサーバから実行時に物理シミュレータを読み込むが、物理シミュレーションに関わるパラメータの多くは定数として定義されているため変更することが出来ない。

^{†1} 電気通信大学
The University of Electro-Communications

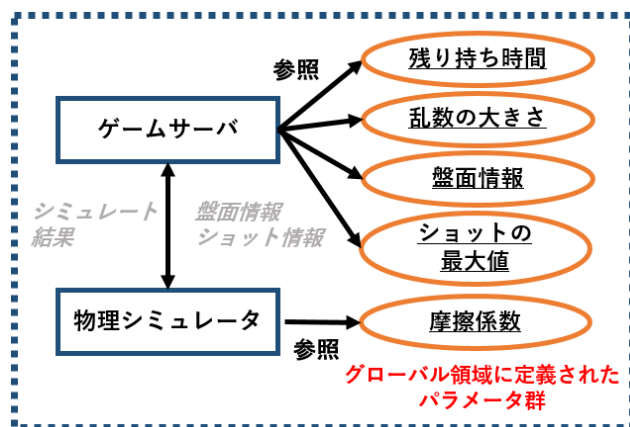


図1 従来システムの構成

2.2 物理シミュレータ

ゲームサーバで用いられる物理シミュレータのライブラリは「Curling Simulator」としてゲームサーバとともに公開されている。このライブラリは DLL 形式で配布されており、ゲームサーバやカーリング AI が実行時に読み込むことで利用することが出来る。本原稿執筆時の最新バージョンは ver.2.3 および大会用に (x,y) 成分にかかる乱数の比率を変更した ver.2.4GAT となっている。

シミュレータ内におけるストーンの大きさや、プレイエリア、ハウスといったリンク内の各エリアの寸法は実際のカーリングに準拠した値となっている。リンク内の各部の座標は、ストーンの進行方向を上としてリンクの左上を原点とし、 x 軸は右方向、 y 軸は下方向を正とする直交座標系で表され、 z 軸方向（リンクに対して垂直方向）については考慮されていない。

2.3 ゲームサーバ

ゲームサーバでは通信プロトコルを介して AI との情報やり取りを行い、物理シミュレータを用いてショットのシミュレーションを行うことで、試合進行の制御を行う。従来システムにおいて試合進行を行う関数は GUI やログの再生機能といった機能を実現する関数とともに、グローバル関数として定義されている。すなわちゲームサーバというものが他の機能と明確に区別されているわけではないが、本論文では便宜上試合進行を行う関数群を従来システムにおけるゲームサーバとして扱う。

ゲームサーバでは、ある局面の情報（ストーン座標、ショット数、エンド数、これまでの得点）に対して、ショットの初速度ベクトル (x,y) の直交座標系で表現される、および曲がり（カール）の方向を与えると、物理シミュレーションが実行され次の局面が生成される。これを各エンド16投分行くと、エンド終了時の得点が自動的に計算される。これを規定エンド数繰り返し、合計得点を計算することでカーリングのゲームをシミュレートする。

2.4 通信プロトコル

ゲームサーバとカーリング AI の通信を行うプロトコルが用意されており、公式サイト上で公開されている[10]。この通信プロトコルでは様々なコマンドが定義されており、試合情報のやり取りに加え、ゲームの開始や終了・カーリング AI の試行開始といった指令の通知に用いられる。

2.5 従来システムの問題点

2.5.1 システム全般

システム全体の問題点として、「それぞれのパラメータが散逸しており参照・変更が困難であること」「カーリングのゲームを表現するのに必要なパラメータが十分に存在しない、もしくは変更不可であること」があげられる。

試合に関するパラメータや物理シミュレーションに関するパラメータがグローバル変数として定義されており、プログラム内で散逸していることから、ユーザによるパラメータの変更が困難となっている。

また、カーリングのゲームには「様々なルールが存在すること」「氷のコンディションが時間ごとに変化すること」「各チーム内でそれぞれのプレイヤーが異なる技量を持つこと」など試合毎に変化する要素を持っている。これがカーリングの戦術を複雑にしているとともに、カーリングのゲームとしての醍醐味でもある。しかし、従来システムではこれらの要素は表現されていないことから、カーリングのゲームを十分に表現したシステムであるとは言い難い。

2.5.2 物理シミュレータ

ショットのシミュレーション

従来システムにおけるショットの曲がり（以下、カールとする）は、ベクトルに対して垂直方向に一定の力がかかるものとして表現されている。しかし現実のカーリングと比較すると、特にストーンが静止する直前のカールが弱いという指摘がされることもあり、すくなくともカールの強さを自由に変更できるように改良する必要がある。また、現状用いられているベクトルに対して一定の力がかかるというモデルの妥当性についても議論の余地がある。

乱数モデル

従来システムの乱数モデルでは、図2のように直交座標系で表したショットの初速度ベクトルの各成分 (x,y) に対して、独立した正規乱数を加えることでショットの不確かさを表現していた。しかしながら現実のカーリングプレイヤーは、「投げる方向」と「投げる強さ」を決めてショットを行っているため、この乱数モデルは不自然であるという指摘がある。

初速度ベクトルをショットの始点を原点とする極座標系で表し、その各成分に対して乱数を加えることで、より現実に近い乱数モデルを実現できるのではないかと考えられる。

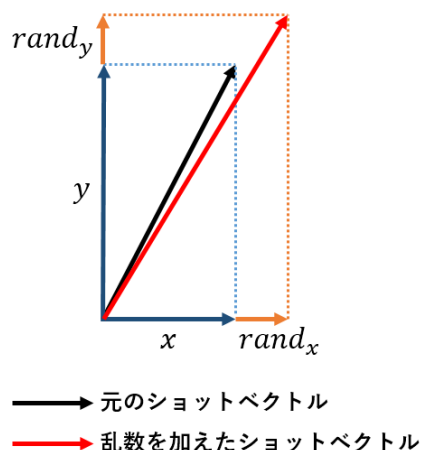


図2 従来システムにおける乱数モデル

2.5.3 ゲームサーバ

条件変更への対応

従来システムにおいて試合条件として「先手・後手の持ち時間」「乱数の大きさ」「エンド数」を指定することが出来る。過去のデジタルカーリング大会では主にこれらの条件を変更することでレギュレーション変更への対応を行っていた。これ以外の変更点については、各種パラメータがコンパイル時定数として定義されているため、これらの数値を変更しビルドしなおしたものを公開することで対応を行ってきた。

例として、2016年11月に開催された第2回GPW杯以降、乱数の大きさはダイアログで指定した数値に対して x 方向が1/2倍、 y 方向が2倍となるように変更されている。この変更に対しては、シミュレータをビルドしなおすことで対応したため、通常の乱数を用いるバイナリと合わせて異なる仕様のシステムが2つ公開されている。大会等で、混乱の原因となっていた。

プレイヤー毎の技量差の表現

カーリングは通常1チーム4人のプレイヤーで構成され、それぞれのプレイヤーが異なる技量を持つ。そこで現実のカーリングの試合ではそれを考慮した上でショットを投げる順番を決定するなど、プレイヤーの技量がチームの戦略に影響を与えていることが知られている。現状のシステムではプレイヤー毎の技量差は考慮されていない。より現実のカーリングに近い条件で試合を行えるシステムが望ましいと考える。

2.5.4 通信プロトコル

従来システムで用いられているプロトコルはシステムの開発初期に策定されたものであり、ルールやレギュレーションの変更を想定していない。そのため、乱数の大きさのような試合条件を通知するコマンドが不足している。

また、現実のカーリングでは得点差が離れ、逆転が難しいことがわかると投了（コンシードと呼ばれる）が行われる。デジタルカーリングではコンシードを行うコマンドが

用意されていないため、逆転が不可能な得点差であっても試合が続行してしまうことがしばしば起こる。大会における時間短縮や、機械学習のために大量の試合を実行する際の効率を考慮しても、コンシードを行うコマンドを追加する必要があると考えられる。

3. 改良システム

3.1 概要

2.5節で述べた旧システムの問題点を踏まえ、物理シミュレータおよびゲームサーバの再構築を行う。再構築した物理シミュレータとゲームサーバを合わせて新たに「Digital Curling」システムと呼称するが、本論文では便宜上、これを改良システムとする。通信プロトコルの見直しも行い、新たに Digital Curling Protocol として定義した。

改良システムの構成を図3に示す。改良システムにおけるゲームサーバは各試合の進行を制御するクラス（試合進行クラス）のインスタンスと、設定ファイルのパスを保持する。試合進行クラスは、いわばカーリングのゲームそのものを表現したクラスであり、試合情報（ルールの種類やエンド数）やシミュレータ、各カーリングAI（本章以降では3.3.3節で述べるチーム内のプレイヤーと区別するため、カーリングAIをクライアントと呼称する）の情報といった、カーリングのゲームの表現に必要なパラメータを全て一括で保持する。

パラメータは試合の実行時に設定ファイルから読み込まれ、試合毎に変更を行うことでルール・レギュレーションの変更に対応する。また、このクラス内で試合の進行を行う処理をまとめた関数群（「試合進行関数群」とする）が定義されており、「試合進行制御部」が試合進行関数を順次呼び出すことで、試合の進行を行う。

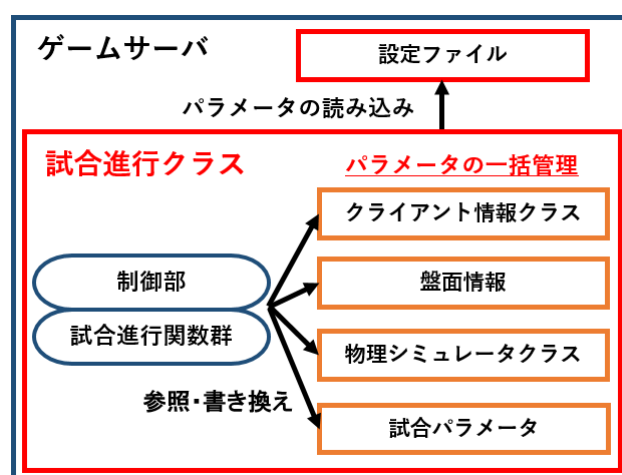


図3 改良システムの構成

3.2 物理シミュレータ

3.2.1 概要

改良システムにおける物理シミュレータライブラリ（以下、改良シミュレータとする）を新たに構築した。ストーン物理運動のシミュレーションに関しては、従来システムの手法を踏襲しつつ、今後のルール・レギュレーション変更を想定し、パラメータの変更に対して柔軟に対応できるように関数の整備を行った。

3.1 で述べたように、改良シミュレータはゲームサーバのビルド時に静的にリンクされる。従来システムを用いた大会においては、公式の物理シミュレータを用いるクライアントは多く存在しているため、改良システムにおいても物理シミュレータを単体で利用することが考えられる。システムのソースコードは公開されているため、改良シミュレータを単体でビルドすることで、静的あるいは動的ライブラリとして利用することが可能である。

3.2.2 ショットのシミュレーション

2.5.2 節で述べたように、デジタルカーリングにおけるストーンの運動モデルの妥当性については疑問が残る。現状ではストーンがカールする物理的な原理が解明されていないことや、実際のストーンが描く軌跡の実測データも十分に存在しないことから、従来システムよりも妥当な運動モデルを構築するのは困難である。

そこで改良シミュレータにおいても同様に速度ベクトルに対して垂直方向の力を加える点や、衝突の処理に 2D 物理シミュレータである Box2D[6]を用いる点など、基本的な運動モデルは従来システムのを踏襲する。なお、使用する Box2D のバージョンは 2019 年時点で最新版である v.2.3.0 とした。

3.2.3 乱数生成器

乱数モデルに関しては、従来システムで用いられていた直交座標系の乱数生成器に加え、極座標系で表した初速度ベクトルの各成分へ乱数を加える乱数生成器を実装した。

シミュレータ内部ではショットベクトルはすべて直交座標系 (v_x, v_y) で表されるため、この乱数生成器ではこれを極座標系 (v, θ) (v : ベクトルの大きさ, θ : 角度) へ変換する。C++ の標準ライブラリを用いて独立した正規乱数 rand_v , rand_θ を生成し、これらを加えたベクトル $(v + \text{rand}_v, \theta + \text{rand}_\theta)$ を直交座標系に変換し実際に投げるベクトルとする。

なお、直交座標系と極座標系どちらの乱数生成器を用いるかは、ゲームサーバ側から設定可能としている。図 4 は、極座標系でショットに乱数を加えたショットベクトルを図示したものである。

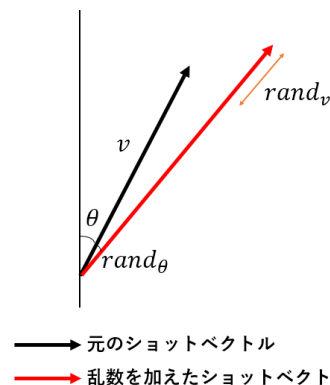


図 4 改良システムにおける乱数モデル

3.3 ゲームサーバ

3.3.1 概要

改良システムにおけるゲームサーバ（「改良サーバ」とする）は、各試合の進行を制御するクラス（便宜上「試合進行クラス」と呼ぶ）を持つ。試合進行クラスは、カーリングのゲームの表現に必要なパラメータを全て一括で保持する。また、試合の進行に必要な処理をまとめた関数群が定義されており、クライアントとの通信やシミュレーションの実行はこれらの関数内で実行される。試合進行制御部と呼ばれる関数がこれらの試合進行関数を順次呼び出すことで、試合の進行を行う。

試合の実行時に試合進行制御部は、JSON 形式で表現された設定ファイルから、試合設定やクライアント情報、物理シミュレータの設定など各種パラメータを読み込む。読み込んだパラメータを用いて試合進行クラスの初期化を行う。規定試合数が終了するまで、試合進行関数を順次呼び出すことで、試合が実行される。

3.3.2 試合進行クラス

試合進行クラス（クラス GameProcess）は、各試合におけるカーリングのゲームを表現したクラスである。試合情報（ルールの種類やエンド数）やシミュレータ（4.2 節のクラス Simulator）・各クライアント（クライアント情報クラス Client）の情報といった、カーリングのゲームの表現に必要なパラメータを全て一括で保持する。

試合進行クラスは試合開始前に、試合毎のパラメータを設定ファイルから読み込む。読み込んだパラメータに基づいてクライアントやシミュレータの初期化を行い、試合の開始をクライアントに通知する。クライアントとの情報のやり取りは、クライアントクラス内で定義された送受信関数を用いる。

試合進行クラスのクラス図を図 5 に示す。試合進行クラスは試合情報やシミュレータ、クライアントの情報をメンバ変数として持つほか、試合の各ステップでの進行処理を行うメンバ関数を持つ。

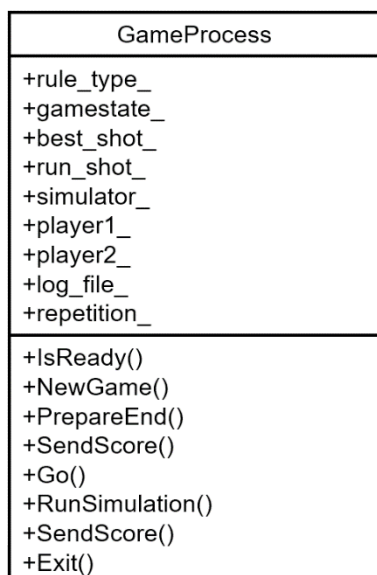


図5 試合進行クラス

3.3.3 プレイヤ毎の技量差の表現

より現実に近いカーリングの試合を再現するために、改良システムでは、従来システムでは存在しなかったプレイヤ毎の技量差を導入する。プレイヤ毎の技量差としては様々な要素が挙げられるが、改良システムではショットを投げる精度と投げられるショット強さをプレイヤの技量差として用いる。

具体的には、初速度ベクトルの各成分にかかる乱数の大きさと、実行可能なショットの初速度ベクトルの最大値を、1チーム最大で4人のプレイヤがそれぞれ異なる値を持つものとして設定ファイルに指定することが出来る(3.3.2節参照)。このパラメータは設定ファイルにおいて配列で記述されており、従来どおりすべてのプレイヤが同一のパラメータを持つようにするには、1人分のパラメータのみを記述すればよい。

試合進行システム内では、プレイヤ毎の技量差は図6のクライアント情報クラス“Client”内で表現される。クライアント情報クラスはクライアントの名前や持ち時間・残り持ち時間に加え、PlayerInfoクラス型のメンバ変数として、最大4人分の乱数の大きさとベクトルの最大値を保持することができる。

チーム内でストーンを投げる順番は、通常ルールで試合毎に、ミックスダブルスルールではエンド毎に変更することが可能である。プレイヤが投げる順番を変更できるように、改良システムでは専用のコマンドの追加を行った。

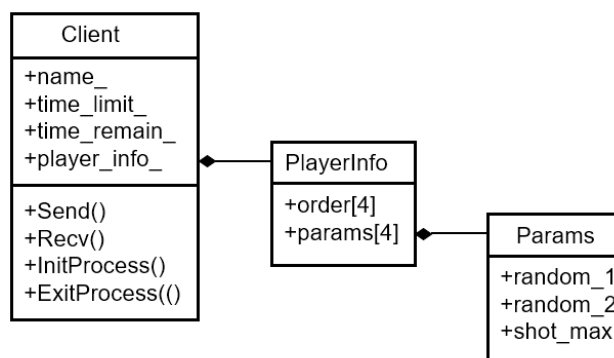


図6 クライアント情報クラス

3.3.4 ミックスダブルスルールへの対応

ミックスダブルスルールはカーリングにおける変則ルールのひとつで、平昌オリンピックより正式種目として採用された。通常のカーリングと異なり、1チームは男性1人、女性1人の2人から構成される。1エンドあたりの投数は各チーム5投・合計10投であり、フリーガードゾーンルールを適用する投数が3投となる。

また、各エンドの最初にあらかじめストーンの配置を決めることができ、この配置がエンドの戦略に大きく影響を与える。ストーンの配置パターンは、ハウスの中央に1つ・センターガード1つ(図7の左:センターガード)およびハウスのサイドに1つ、サイドガード1つ(図7の右:パワープレイ)の2種類、またどちらのチームのストーンを手前にするかという組み合わせを考えると合計4種類が存在する。ストーンの配置は前のエンドを失った(得点できなかったチーム)が決定し、各チームパワープレイは1試合に1度しか指定することができない。

改良システムでミックスダブルスルールの試合を実行するために、各エンドのはじめにストーンの配置を行うコマンドを用意した。また投数(GameState.ShotNum)は6から始まり、15が最終投とすることで、1エンド10投のゲームを再現した。

現実のカーリングにおいてストーンが配置される具体的な位置は、各試合で氷のコンディションなどを考慮して決定される。位置の決定はアイスメーカーと呼ばれるスタッフの裁量によって変わることから、改良システムにおける配置座標は、カーリングのルールブック[7]に記載されている典型的な値を用いることとする(図3-5)。

このようにミックスダブルスは通常ルールとは大きく異なるため、どのルールで試合が行われるかは、ゲーム情報の通知を行う「GAMEINFO」コマンドによって各カーリングAIへ通知される。

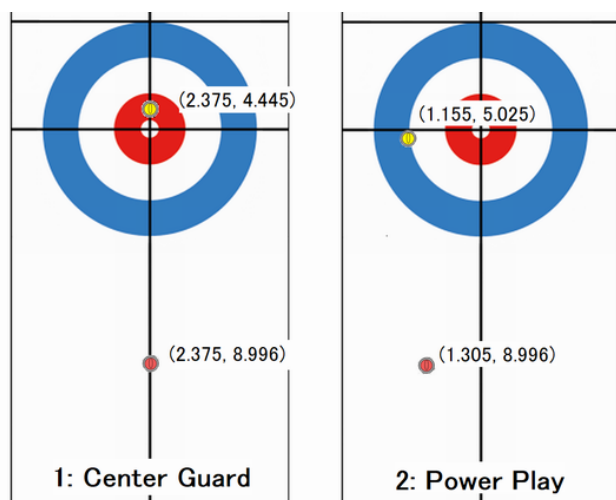


図7 ミックスダブルスにおけるストーンの初期配置

3.4 通信プロトコル

従来システムの問題点を踏まえ、改良システムでは従来のプロトコルを拡張した Digital Curling Protocol (以下、DCP) の定義を行なった。従来のプロトコルと異なる点として、各種情報（試合設定・プレイヤーのパラメータ）を通知するコマンド・プレイヤーの投げる順番を指定するコマンド・ストーンの配置を行うコマンド・コンシードを行うコマンドが追加されている点が挙げられる。

- ・ GAMEINFO rule random num players
試合情報を通知するコマンド
 - “rule”：ルールの種類（0：通常，1：ミックスダブルス）
 - “random”：乱数生成器の種類（0：直交座標系，1：極座標系）
 - “num_players”：1チームあたりのプレイヤーの数
- ・ PLAYERINFO rand1_0 rand2_0 shotmax_0 ... rand1_3 rand2_3 shotmax_3
プレイヤーのパラメータを通知するコマンド
 - “rand1_n”：n番目のプレイヤーの乱数の大きさ1
 - “rand2_n”：n番目のプレイヤーの乱数の大きさ2
 - “shotmax_n”：n番目のプレイヤーのショットの上限値
- ・ SETORDER
- ・ SETORDER p1 ... p3
プレイヤーの投げる順番を指定するコマンド

- “pm”：m番目に投げるプレイヤーの番号
(PLAYERINFOで通知された順番)

- ・ PUTSTONE（ゲームサーバ→クライアント）
- ・ PUTSTONE type（クライアント→ゲームサーバ）
ストーンの配置を指定するコマンド
 - “type”：ストーンの配置
(0：センターガード，配置プレイヤーが後手)
(1：センターガード，配置プレイヤーが先手)
(2：パワープレイ，配置プレイヤーが後手)
(3：パワープレイ，配置プレイヤーが先手)

3.5 改良システムの応用

3.5.1 改良システムを用いた大会の実施

改良システムを用いたデジタルカーリング大会を、第23回ゲームプログラミングワークショップにおいて GPW 杯 2018 デジタルカーリング大会という形で開催した。大会は従来システムで行われたものと同形式の部門を通常部門とし、改良システムによる条件変更の一例として新にミックスダブルス部門を設け、2部門それぞれについて優勝を決定した。

大会は大きなトラブルなく進行し、ルール・レギュレーションの変更を伴う大会において改良システムは問題なく動作することを確認した。

3.5.2 戦略学習支援システムへの応用

改良システムを樋口らが開発したカーリングの戦略学習支援システム[8]へ応用することも考えられる。この学習支援システムではストーン配置や得点の編集を行いカーリング AI に思考させることができるが、従来システムを踏襲しているため物理シミュレータのパラメータ変更やプレイヤーの技量差といった要素には対応していない。ストーンの曲がりの強さやショットの精度（乱数の大きさ）・強さといった要素は、カーリング AI や実際のカーリングプレイヤーの戦略の影響を与える要素である。

例として、図8に学習支援システムにおいて、異なるショットの強さの上限値を持つ2つのAIが思考したショットを示す。従来大会と同等の上限値を持つAIの結果が赤、より大きな上限値を持つAIの強さが緑で示される。標準的な上限を持つAIは相手のストーンに軽く当たってのショットを選択しているのに対して、上限値の大きいAIは、相手のストーンをハウス内からはじき出す強力なショットを選択している。このように、ショットの強さのようなパラメータはカーリングAIの戦略に大きく影響を与えることがわかる。

この例ではカーリングAIのソースコードを直接書き換えることで、ショットの強さの変更に対応している。本研

究で構築した改良システムを用いることで、ソースコードの改変を伴わずに、このようなパラメータ変更に対応することが可能となる。実際に学習支援に用いる際は、ユーザ（プレイヤー）の技量に応じて様々な値を設定することになるため、今後の学習支援システムの発展に際して、改良システムの柔軟性は大変有用であると考えられる。

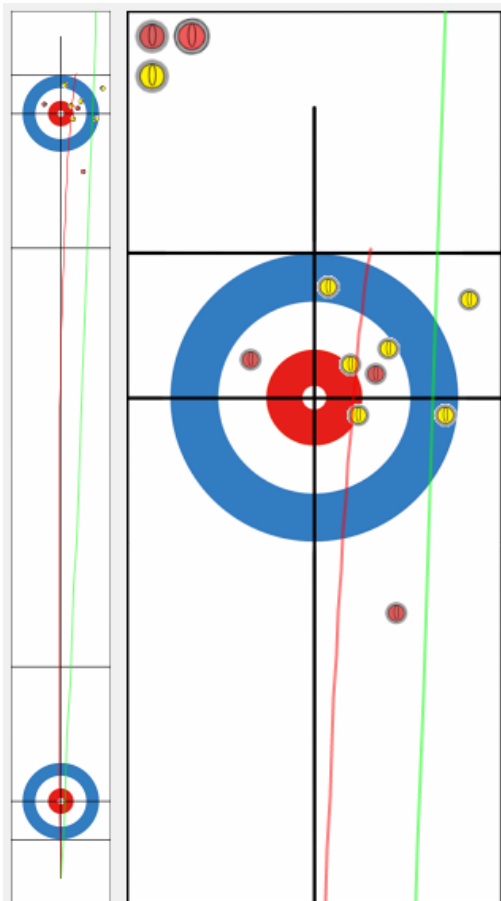


図8 ショットの上限值を変えた場合の思考結果

4. おわりに

本研究では、従来のデジタルカーリングシステムの問題点を踏まえ、ルールや物理シミュレーションのパラメータの変更に対して柔軟に対応できる改良システムを新たに構築した。従来システムでは散逸していた上に、変更できなかったパラメータをひとまとめにし、ゲーム進行に必要な関数群も合わせたゲーム進行システムを定義した。これにより、ゲームサーバからルールだけでなく物理シミュレーションに必要なパラメータ等の変更も行えるようになった。また従来システムで表現されていなかったプレイヤー毎の技量差を、乱数の大きさとショットベクトルの最大値という形で表現できることを示した。

本システムを用いた GPW 杯 2018 大会を、通常ルールに加えカーリングの変則ルールであるミックスダブルスにて実施し、ルールの変更に対して柔軟な対応が可能であるこ

とも確認した。レギュレーションの告知を行う時間が十分に取れなかったため、プレイヤー毎の技量差についてはこの大会では導入を見送ったが、2019 年 3 月に GAT2018 (Game AI Tournament 2018[13]) において実施予定の第 5 回 UEC 杯 デジタルカーリング大会では、ミックスダブルス部門において異なるパラメータを持つ 2 人のプレイヤーを想定し導入を行う。

変則ルールや技量差の導入を行い、より現実に近い複雑なゲームを実現することが可能となったことで、改良システムがカーリング AI 研究のより一層の発展に寄与することを期待する。

5. 今後の展望

本研究で構築した改良システムによって、従来システムの問題点であった「現実のカーリングを十分に表現できていない点」について改善を行った。しかしながら、「氷のコンディションの時間変化」や「スウィーピング」といった改良システムにおいても表現されていない要素は残されている。

これらの要素の導入を見送った理由として、カーリングにおける石の物理運動は原理が解明されていない点が多いことが挙げられる。一般的に氷は時間とともに解けていくため摩擦係数が増大し、石が滑りにくくなるだけでなく、石が通過した部分も氷が解けるため、氷の状態は局所的にも変化することが知られている。しかし具体的に石の運動にどのように影響を与えているかについては詳細には分かっておらず、精度良く測定された軌跡のデータも存在しない。スウィーピングについても同様に、石の運動にどの程度影響するか数値化するのは現状では困難である。

しかし現在、実測データを用いて石の運動を数値化する試みが行われている。本論文の執筆時点では具体的な数値の計測には至っていないが、このデータを用いてデジタルカーリングにおける石の運動モデルを新たに構築することで、より現実に近いゲームの再現を行える可能性がある。これにあわせて氷の時間変化やスウィーピングの影響についてもデータが得られれば、デジタルカーリング上でのモデル化や実装も可能であると考えられる。

謝辞 本研究は JSPS 科研費 15H02797 の助成を受けたものである。

参考文献

- [1] Takeshi Ito and, Yuuma Kitasei, Proposal and Implementation of “Digital Curling”, 2015 IEEE Conference on Computational Intelligence and Games (CIG2015), pp.469-473 (2015).
- [2] 北清勇磨, 岡田雷太, 伊藤毅志, デジタルカーリングサーバーの提案と紹介, 情報処理学会ゲーム情報学研究会報告, GI-

- 31(2) (2014).
- [3] 伊藤毅志, 森健太郎, デジタルカーリング大会報告 2015 年度, エンタテインメントコンピューティング, 2016-EC-41, pp.1-6 (2016).
 - [4] 加藤修, 飯塚博行, 山本雅人, 不確定性を含むデジタルカーリングにおけるゲーム木探索, 情報処理学会論文誌, 57(11), pp.2354-2364 (2016)
 - [5] 大渡勝己, 田中哲郎, カーリング AI に対するモンテカルロ木探索の適用, ゲームプログラミングワークショップ 2016 論文集, 2016, pp.180-187 (2016)
 - [6] Box2D, 入手先(<https://box2d.org/>) (参照 2019-1-21).
 - [7] World Curling Federation - Rules and Regulations, 入手先(<http://www.worldcurling.org/rules-and-regulations-downloads>) (参照 2019-1-25)
 - [8] Toshiyuki Higuchi, Kentaro Mori, Takeshi Ito, An AI-Assisted Strategy Learning Support System Using Digital Curling, WCI2018 論文集, pp.21-25 (2018)
 - [9] GAT@UEC 入手先(http://minerva.cs.uec.ac.jp/gat_uec/wiki.cgi) (参照 2019-1-25)