玉入れにおける最適な動きとは
～強化学習による「玉入れＡＩ」の作成と分析～

川越在人　新井寿松　石井佳大
群馬県立高崎高等学校

要旨

玉入れにおける最適な動きについて調べるために、個人戦略（個人として、どこからどのように投げれば最も入りやすいのか）とチーム戦略（チームとして、人の配置や役割分担をどのようにすればより多くの玉が入るか）について、その効率的な動きを考察した。個人戦略では、Unityの物理エンジンを使用したシミュレーションを行った。チーム戦略では、「Unity ML-Agents」による強化学習を用いて「玉入れＡＩ」を自作し、実際の人による玉入れの動きと比較した。結果として、チームとして人間らしい挙動で玉入れを行うことができる、おそらく世界初となる「玉入れＡＩ」を作成することに成功した。個人戦略としては、かごからの距離が1.5m付近から投げるのが最も安定して多くの玉が入ることという結論を得た。チーム戦略では、多くの人をカゴの近くに集め、2、3人は端に寄らせて玉を中心に寄せる役割を担わせるのが良いこと、また、6個の玉を集めてから投げるという玉入れの定石とは異なり、同時に投げる玉の個数に制約を設けずに、状況に応じて柔軟に投げる玉の個数を判断した方が、多くの玉が入る可能性があるという結論を得た。

1. はじめに

1.1　研究の目的

玉入れは運動会の競技の１つとして、幅広い年代に親しまれており、国際的な運動会イベントでも競技の１つとして実施されている。高崎高校（本校）と群馬県立前橋高校は毎年10月に75年間も続いている「定期戦」と呼ばれる競技会を実施しているが、その中でも玉入れは配点が最も大きい競技の１つであり、緻密な戦略とチームワークによる真剣勝負が行われる。玉入れの先行研究は調べてみてもほとんどなく、見つかったのは「玉入れの玉の散らばり具合の抑制」[1)]という研究のみだった。その研究は、6個の玉を集めてから投げるという玉入れの定石に対して、6つ同時に投げる際に玉の積み方によって散らばり具合がどのように変化するかについて実験したもので、人の動きに着目したものは見当たらなかった。そこで、玉入れにおける人の効率的な動きについて研究することで、私たちにとって重要な定期戦における玉入れ競技を有利に進めることができるとともに、現在、経験則に基づいて行われている玉入れの戦略に、根拠に基づいた知見を与えることができるのではないかと考えた。

玉入れでは人の動きとして、個人戦略（個人として、どこからどのように投げれば最も入りやすいのか）とチーム戦略（チームとして、人の配置や投げ方、役割分担をどのようにすればより多くの玉が入るか）が重要である。本研究の目的は、玉入れにおける個人戦略とチーム戦略について、その効率的な動きを考察することである。

1.2　研究の概要

高崎高校と前橋高校で行われる「定期戦」における玉入れのルールを採用した。個人戦略では、Unityの物理エンジンを使用したシミュレーションを行った。チーム戦略としてUnityとGoogle colaboratoryを用いて、Unity ML-Agentsによる強化学習による「玉入れＡＩ」を自作し、実際の人による玉入れの動きと比較した。「玉入れＡＩ」の作成は先行研究が見当たらず、おそらく世界初となる試みである。
ソースコードや実験動画については参考文献＊にリンクを記載する。

1.3　仮説

定期戦で高崎高校や前橋高校が実際に行っている戦略から考えると、個人戦略としては、カゴの近くから投げるほうが入りやすく、チーム戦略としては、6個集めてから投げたり、何人かはボールを集めることに注力したりする方がチーム全体としてはより多くの玉が入るだろう。

2. 本研究における用語の定義

(1)定期戦ルール

高崎高校と前橋高校で行われる「定期戦」における玉入れのルールのことである。図１に定期戦における玉入れの様子を示す。ルールの詳細は次のとおりである。

・カゴの高さ：4.30m　・カゴの直径：0.36m　・カゴの深さ：0.38m
・コートの直径：7.0m　・1ゲームあたりに使用される玉の数150個
・参加人数は15人

ただし、このルールは全日本玉入れ協会の競技ルールとは異なる。

図１　定期戦における玉入れの様子

(2)Unity

Unityとは、Unity Technologiesが開発・販売しているゲームエンジン、物理エンジンのソフトである。本研究では、物理エンジンを用いたシミュレーションや「Unity ML-Agents」による「玉入れＡＩ」の推論とその結果となる人や玉に見立てたモデルの動きの可視化に使用した。

(3)Google colaboratory

Google colaboratoryとは、Googleのクラウドサーバーでコードを実行できる環境である。コードの実行にGPU（Graphics Processing Unit）を使用することが可能であり、本研究では、「Unity ML-Agents」の「玉入れＡＩ」の学習に使用した。

3. 実験１「個人戦略」

3.1　目的

「個人として、どこからどのように投げれば最も入りやすいのか」について明らかにする。

3.2　方法

物理エンジンのUnityを用いて玉入れのモデルを作り、人の身長の30cm上から、玉に初速度を与え、斜方投射させるシミュレーションを行う。図２にシミュレーションの様子を示す。初速度は、かごに最もよく入る初速度を計算しておき、それを1.0倍として、乱数による誤差（0.9倍～1.1倍）をつける。身長、人とかごとの距離、投射角度をそれぞれ変えて100個の玉を3回投げ、その平均を記録する。

＜シミュレーションの条件＞

(1)固定する条件

・かごの大きさ 4.30m ・かごの直径 0.36m・かごの深さ 0.38m ・コートの直径 7.0m
・空気抵抗なし

(2)変化させる条件

・身長 160cmと170cm ・投射角度 65度～89度（１度ずつ変化）
・人とかごとの距離 0.5m～3.5m(0.5mずつ変化)

(3)乱数

・初速度につける誤差範囲 0.9倍～1.1倍
（かごに最もよく入る初速を計算しておき、それを1.0倍とする）

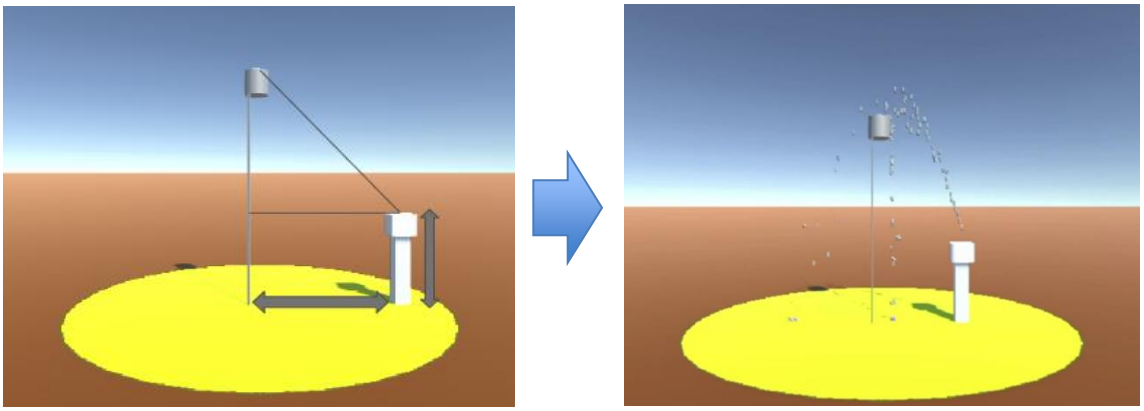


図２　Ｕｎｉｔｙによるシミュレーションの様子
（参考文献＊に動画へのリンクを記載）

3.3　結果

図３に身長160cmの場合（190ｃｍから投射）と身長170ｃｍの場合（200cmから投射）の結果を示す。

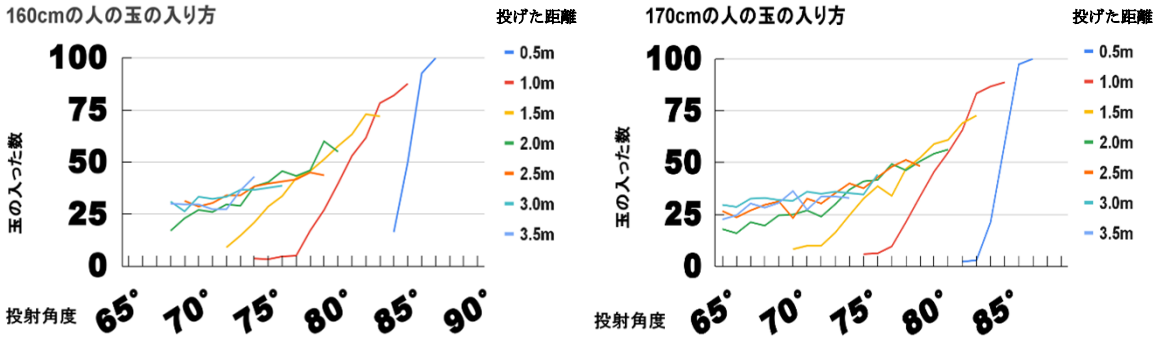


図３　シミュレーションの結果

身長によらず、かごからの距離が1.5m（黄色）の場合には、角度が変化してもバランスよく入ることがわかる。かごからの距離が近い0.5m（青色）、1.0m（橙色）の場合には、玉が入る最大値は大きいが、わずかな角度の変化で入る数が大きく減少する。また、かごからの距離が遠い2.0m（緑色）以上の場合には、どのような角度であっても玉が入りにくくなることがわかる。

3.4　考察

図４にシミュレーションの考察を示す。斜方投射の軌跡に基づいて考えると、かごから遠いと、小さな初速のずれで入らなくなってしまい、距離が近すぎると、小さな角度のずれで入らなくなってしまうと考えられる。**結果として、かごからの距離が1.5m付近から投げるのが最も安定して多くの玉が入ると考えられる。**

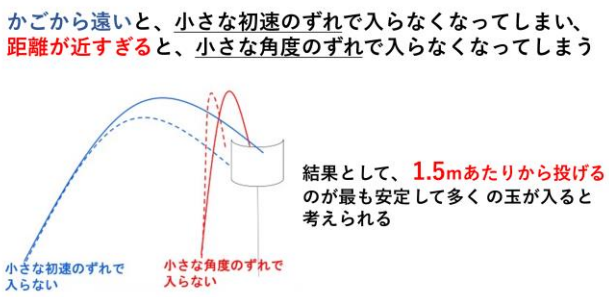


図４　シミュレーションの考察

4. 実験２「チーム戦略」

4.1　目的

「チームとして、人の配置や投げ方、役割分担をどのようにすればより多くの玉が入るか」を明らかにする。

4.2　方法

玉入れに参加する15人を動かして、多く玉が入るように最適化を行う「玉入れＡＩ」を様々な条件で作成し、分析する。このＡＩの作成は、大量のデータの中から規則性を見つけ、分類や判断といった推論のためのルールを機械に生成させる手法である機械学習の一つである「強化学習」という手法で行った。また、Unityで機械学習の学習環境を構築するためのフレームワークである「Unity ML-Agents (Unity Machine Learning Agents)」[2)]を用いた。

4.2.1　強化学習

強化学習は、表１のように用語を定義すると、「エージェント」が「環境」の「状態」に応じてどのように「行動」すれば「報酬」を多くもらえるかを求める手法である。エージェントは、「状態」に応じて次の「行動」を決定するための戦略、つまり「ポリシー」を、より多くの「報酬」を得られるものにしていくのが目的である。そのために、「収益」を最大化する必要があるが、「収益」はまだ発生していない未来の出来事で不確定のことであるので、現在の「状態」から計算される条件付きの「収益」である「価値」が大きくなる条件を探せばよい。つまり、「価値の最大化」が「収益の最大化」につながり、さらに「多くの報酬がもらえるポリシー」という強化学習の目的につながるということである。

表１ 強化学習の用語

用語	説明
エージェント	環境に対して行動を起こす主体
環境	エージェントがいる世界
行動	エージェントがある状態において採ることができる行動
状態	エージェントが行動に応じて更新される環境が保持する状態
報酬	エージェントの行動に対する環境からの評価
ポリシー	エージェントが行動を決定する原理
即時報酬	行動直後に発生する報酬
遅延報酬	遅れて発生する報酬
収益	即時報酬だけでなく、後に得られるすべての遅延報酬を含めた報酬和
価値	エージェントの状態とポリシーを固定した場合の条件付き収益

4.2.2　学習サイクル

図５に学習サイクルを示す。「次の状態」が現在の「状態」と「行動」によって確定するシステムである「マルコフ決定過程」という強化学習サイクルで行う。具体的に「玉入れＡＩ」を用いて説明する。ただし、「玉入れＡＩ」においてPlayerとは、玉入れに参加する選手、「エージェント」は15人のPlayerを操作する主体（１つのＡＩが15人を操作）、「環境」は、9m×9mのグラウンド上で実験①と同じ定期戦ルールに合わせた環境とする。またPlayerが玉を拾う距離は半径20cm、玉を他のPlayerに渡せる距離は半径50cmとした。

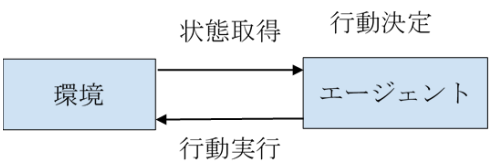


図５　学習サイクル

(1)状態取得

表２にＡＩの学習設定を示す。表２の設定で「エージェント」は「環境」に対して観察を行い、「状態」を取得する。表２の「Grid Sensor」とは、コートを8100等分（90×90）し、コートに存在している物体（人、玉）を認識するセンサーのことである。図６に「Grid Sensor」の様子を示す。また、表２の「Vector Observation」はPlayerが玉を持っている数などを浮動小数配列として観察するデータである。

表２ AIの学習設定

観察	gridsensor （人や玉の位置） 90×90 Vector Observation（サイズ45） 0～14：Playerの玉を持っている数(整数) 15～44：Playerのxz移動速度
行動	Continuous(サイズ30) 0,1：Playerのxz移動速度×15人 Discrete（サイズ30） 0～14：投げるかどうか(0:投げない、1:投げる)×15人 15～29：渡すかどうか（自分以外の人に1対1で自分の持っている玉すべてを渡す）(0:渡さない、1～15:それぞれに対応している人)
報酬	玉を拾ったら +0.01 玉が入ったら +0.1×(玉の個数) ステップ毎 -1/MaxStep 150個の玉が全て入ったら +30
決定	5ステップ毎　MaxStep：5000

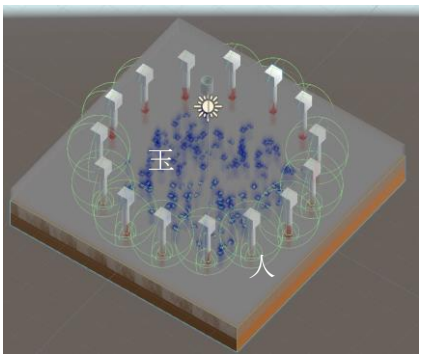


図６　「Grid Seosor」の様子

(2)行動決定・実行

「エージェント」が持つ「ポリシー」が観察した「状態」に応じて「行動」を決定し、「エージェント」はその「行動」を実行する。「Unity ML-Agents」では、-1.0～1.0の連続値の要素を持つ浮動小数配列を利用した「Continuous」な「行動」と、0,1,2…のような離散値の要素をもつ整数配列を利用した「Discrete」な「行動」の２つを利用できるため、「玉入れＡＩ」では表２の「行動」を行えるようにした。ただし、学習の効率を上げるために、球を投げられないときは、表２の「1:投げる」を選択できないようにするなど、行動マスクを行った。

(3)報酬取得・ポリシー更新

「エージェント」は「行動」の実行結果に応じて「報酬」を受け取る。そして、どのような「状態」でどのような「行動」を取ったら、どの程度「報酬」がもらえたかという「経験」に応じて、「ポリシー」を更新する。学習アルゴリズムはPPO（Proximal Policy Optimization）を使用した。PPOは、「エージェント」が持つ「経験」を用いて、ほかの学習アルゴリズムより安定して「ポリシー」を更新することができるアルゴリズムである[3)]。「玉入れＡＩ」では、表２のように「報酬」を設定した。

(4) ポリシーの決定

(1)から(3)を繰り返して将来的に多くの報酬を得られるポリシーを求めた。強化学習の学習1回分を「1エピソード」、学習環境の1フレームを「1ステップ」と呼ぶが、「玉入れＡＩ」では、エピソード開始からエピソードの最大ステップ数であるMaxStepとなるまでか、Playerが落下した時点のステップまでを1エピソードとし、そのエピソード内でもらえる報酬が最大になるように「ポリシー」を更新していく。また、Unityでは基本的に0.02秒毎で呼ばれるFixedUpdate()というメソッドが呼ばれるタイミ

ングを「1ステップ」としている[4)]。そのため、(1)から(3)のサイクルが行われる「決定」の時間間隔を、人間の反応時間の限界といわれる約0.1秒[5)]を考慮し、0.1秒/0.02秒＝5ステップに設定した。

4.2.3　学習と推論

Unity上でＡＩの学習設定を行い、設定ファイルを出力した。ＡＩの学習は、Google colaboratoryに設定ファイルをアップロードし、オンライン環境でGPUを利用して行った。ＡＩの推論はUnityを用いて行った。「玉入れＡＩ」はボールを最大6個まで持つことができるようにし、その数を0個が白、1個が黄緑、2個が水色、3個が青、4個が紫色、5個が桃色、6個が赤色の頭になるようにしてUnity上で可視化した。

4.3　結果および考察

4.3.1　分析Ⅰ「玉入れＡＩ」の人と玉の配置の時間変化

＜結果＞

図７に、玉を投げる初速度をかごに最もよく入る最適化した値v_0から、誤差として$\Delta v_0=\pm 0.1v_0$を設定し、かつ、玉を1個でも持ったら投げても良い（最大で6個持てる）という条件で学習したＡＩの推論結果を示す。実際には動画になっており、「玉入れＡＩ」が人間のように、玉を拾ったり、投げたり、近くの人に玉を渡したりする様子が観察できた。

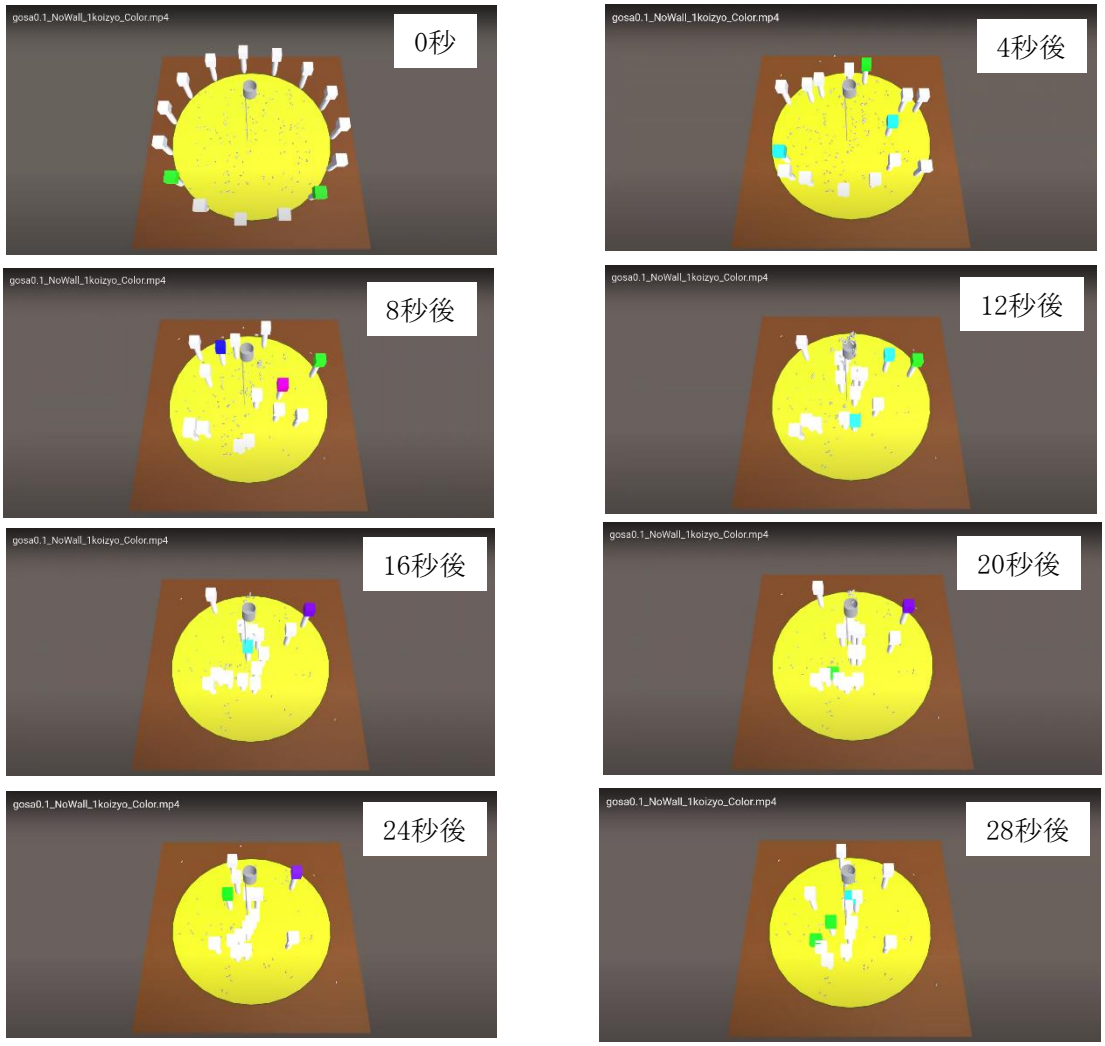


図７　AIの推論結果の可視化の様子

（参考文献＊に動画へのリンクを記載）

人と玉の配置の時間変化を観察すると、開始直後は、玉を持ったら遠くからでも投げている（4秒、8秒後）。結果として、入らなかった玉がかごの近くに集まり、その玉を拾うために人もかごの近くに集まってくる（12秒後以降）。かごの近くでは、頻繁に玉の受け渡しが行われ、玉を1個～3個持った段階で玉を投げている様子が観察できた。また、2～3人はかごから遠くに配置されていることがわかる。

＜考察＞

「玉入れＡＩ」で見られた人の配置や役割分担は実際の玉入れでも観察される。図８に実際の玉入れと「玉入れＡＩ」との比較を示す。プレーヤーのほとんどはかごの近くに集まり、玉をかごに入れる役割を担い、一部のプレーヤーは遠くで、玉を拾ったり、かごの近くに玉を集めたりする役割を担っている。「玉入れＡＩ」では、実験１の個人戦略の結果から、かごから1.5m付近が最も入りやすく、この付近で玉を投げた方が多くの報酬を得られることから、このようなチーム戦略をＡＩが学習していると考えられる。以上のことから、**チーム戦略としては、多くの人をカゴの近くに集め、2、3人は端に寄らせて玉を中心に寄せる役割を担わせるのが良い**と考えられる。報酬を与えることで学習した「玉入れＡＩ」と経験則から学習・行動している人間で、同じチーム戦略をとるのは興味深い。

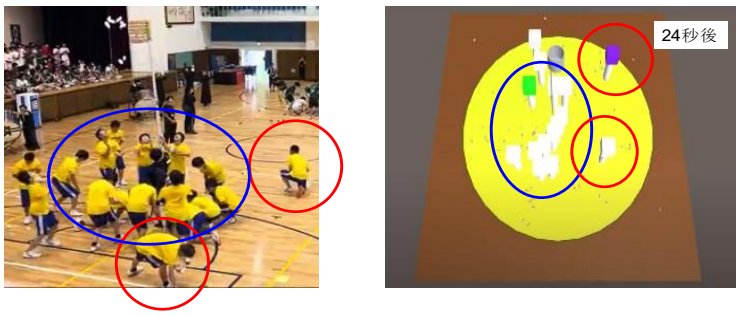


図８　実際の玉入れと「玉入れAI」との比較

4.3.2　分析Ⅱ「6個持ってから投げるという定石は戦略的に優れているか」

＜結果＞

次に、条件Ｉのように1個でも持ったら投げてもよいという条件と、玉入れの定石である6個持ってから投げる戦略のどちらが優れているのかを確かめるために、玉を6個持ってから投げる（⑥と表記）と、1個でも持ったら投げてもよい（①以上と表記）という2つの条件でＡＩの学習を行った。比較のため、玉を投げる初速度をかごに最もよく入る最適化した値v_0から、$\Delta v_0=\pm 0.1v_0$（誤差小と表記）と$\Delta v_0=\pm 0.55v_0$（誤差大と表記）で学習を行った。図８に学習経過の様子としてステップ数ごとの報酬を示す。

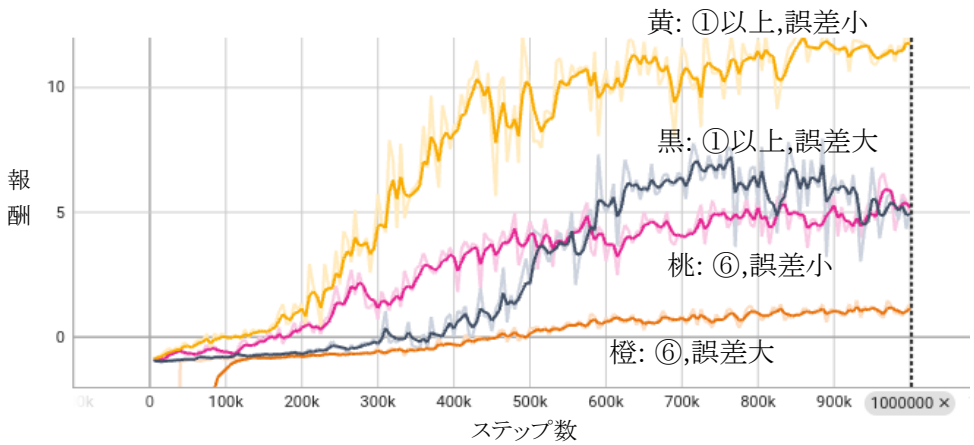


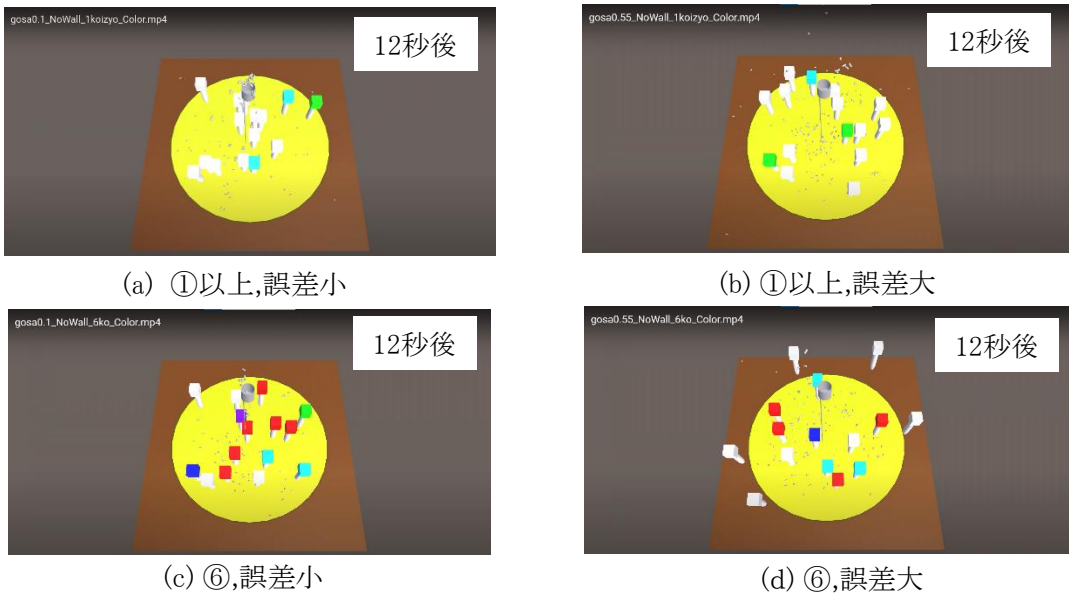
図8　学習経過の様子

（黄:①以上,誤差小　黒: ①以上,誤差大　桃: ⑥,誤差小　橙:⑥,誤差大）

最終報酬は、「①以上,誤差小」が11.69、「①以上,誤差大」が4.938、「⑥,誤差小」が5.198、「⑥,誤差大」が1.148であった。同じ誤差で比較すると「⑥」より「①以上」の方が約2倍多く報酬を獲得している。

＜考察＞

最終報酬の結果から、定石とは異なり、玉を6個集めるのを待たずに投げた方がより多くの玉を入れられると考えられる。この理由を考察する。図10に、「玉入れＡＩ」の開始から１２秒経過したときの推論結果の比較を示す。



(a) ①以上,誤差小　(b) ①以上,誤差大

(c) ⑥,誤差小　(d) ⑥,誤差大

図10 「玉入れAI」の推論結果の比較

図10で、かごの近くに集まっている人の割合を比べると、(a)＞(c)≒(b)＞(d)となっており、最終報酬が低いものほど、人がかごの近くに集まらず、かごから広がって配置されていることがわかる。「1個以上で投げる」という戦略の場合、投げる回数が多く、かごに入らなかったとしても、より早くかごの近くに玉が集まっていた。その結果、かごの近くに人もより早く集合していた。かごの近くで投げた方が、玉が入りやすいため、多くの玉がかごに入ると考えられる。逆に「6個持ってから投げる」という戦略では、玉を6個集めるのに時間がかかり、玉がかごの近くに集まらず、玉が入りにくくなると考えられる。また、分析Ⅰの結果では、人がかごの近くに来たとしても、最大で6個まで持てるにも関わらず、1個～3個で玉を投げていた。以上のことから、少なくとも今回のAIの条件設定においては、**定石とは異なり、玉を6個集めるのを待たずに投げた方がより多くの玉を入れられる**と考えられる。

一方で、玉を投げられる最小の個数が何個の場合が最も玉が入るのか、また、そもそも玉を投げる個数を固定しないほうが良いのかという疑問が生じた。そこで、玉入れにおける最適な制約条件を調べるために分析Ⅲを行った。

4.3.3　分析Ⅲ「玉入れにおける最適な制約条件はどのようなものか」

＜結果＞

分析Ⅱと同様の環境で、初速度の誤差を「誤差小（$\Delta v_0 = \pm 0.1 v_0$）」として、2個以上、３個以上、５個以上で投げることが可能となる「②以上,誤差小」「③以上,誤差小」「④以上,誤差小」「⑤以上,誤差小」の4つと、３個持ったときのみ投げることが可能な「③，誤差小」の新に５種類のＡＩの学習を行った。図11に学習経過の様子としてステップ数ごとの報酬を示す。

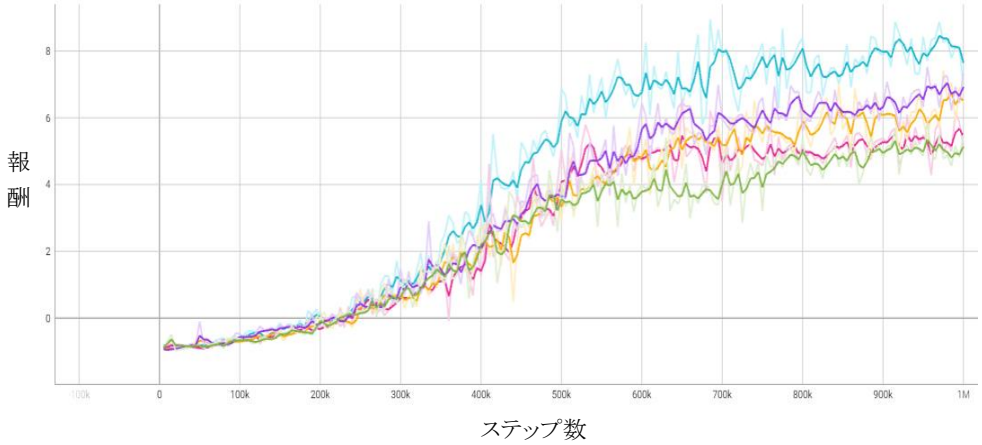


図9　学習経過の様子

（水色: ②以上,誤差小　紫: ③以上,誤差小　黄: ④以上,誤差小　緑:⑤以上,誤差小　桃: ③,誤差小）

最終報酬は「②以上,誤差小」が7.64、「③以上,誤差小」が6.918、「④以上,誤差小」が6.519、「⑤以上,誤差小」が5.129であり、「③,誤差小」が5.438であった。

＜考察＞

分析Ⅱ「①以上,誤差小」の最終報酬が11.69であったことも考慮すると、球を投げる最小の個数の条件が少ないほど、最終報酬が高くなっている。また、「③,誤差小」よりも「③以上,誤差小」の方が、最終報酬が高かった。これらのことから、**同時に投げる玉の個数に制約を設けずに、状況に応じて柔軟に投げる玉の個数を判断した方が、多くの玉が入る**と考えられる。

5. まとめ

5.1　結論

「UnityML-Agents」による強化学習を用いて、チームとして人間らしい挙動で玉入れを行うことができる、**おそらく世界初となる「玉入れＡＩ」を作成することに成功**した。

個人戦略としては、1.5m付近から投げるのが最も効率が良く、また、チーム戦略としては多くの人をカゴの近くに集め、2、3人は端に寄らせて玉を中心に寄せる役割を担わせるのが良いという結論を得た。また、今回の条件で学習した「玉入れＡＩ」においては、６個玉を集めてから投げるという玉入れの定石とは異なり、同時に投げる玉の個数に制約を設けずに、状況に応じて柔軟に投げる玉の個数を判断した方が、多くの玉が入る可能性があるという結論を得た。

本研究の成果は、玉入れの戦略に対して、強化学習によるＡＩの作成と分析という新たなアプローチを可能とするものである。

5.2　今後の展望

玉入れの定石とは異なる結果が得られた点について、現実の玉入れにおいてもそのような戦略を用いるのが最適である可能性があるため、実際の試合において検証をする必要がある。「玉を投げる」という行動にかかる時間が人間よりも短いことなど、実際の人間の動きを完全に再現しきれていないため、より詳細な条件について考慮すべきである。また、最適な戦略は、ＡＩを学習する際の報酬などのパラメータによって変わると考えられるので、様々なパラメータでさらなる検証が必要である。本研究では「玉入れＡＩ」の作成に関して、より安定して学習が行える「1つのＡＩが15人を操作する手法」を用いたが、15人それぞれに別のＡＩを持たせるマルチエージェントにすることで、どのような戦略が生み出されるのかも興味深い。

謝辞

指導教員である群馬県立高崎高等学校の岡田直之先生には論文執筆における助言をしていただきました。深く感謝致します。

参考文献

＊本研究のソースコードや実験動画
https://github.com/Kawato777/ThrowingBalls/releases/tag/v1.0.0

1)飯島拓真ら（2020）「玉入れの玉の散らばりの抑制」群馬県立高崎高等学校論文集
2)布留川英一　(2022)「Unity ML-Agents 実践ゲームプログラミング v2.2対応版」ボーンデジタル
3)Proximal Policy Optimization – Spinning Up documentation(https://spinningup.openＡＩ.com/en/latest/algorithms/ppo.html)　2024年6月閲覧
4)MonoBehaviour-FixedUpdate() - Unity スクリプトリファレンス(https://docs.unity3d.com/ja/2021.1/ScriptReference/MonoBehaviour.FixedUpdate.html)　2024年6月閲覧
5)猪飼道夫ら（1960-1970）「全身反応時間の研究とその応用」財団法人日本体育協会(https://www.japan-sports.or.jp/Portals/0/data/supoken/doc/studiesreports/1960_1970/S3809.pdf)