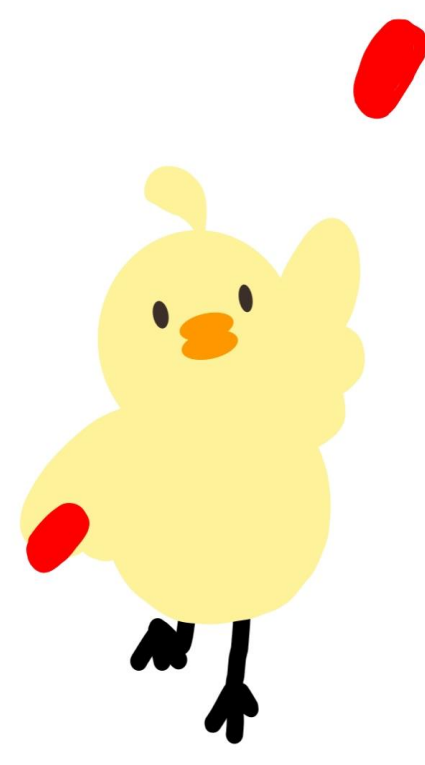


玉入れの最適な動きとは～物理エンジンと機械学習による分析～

群馬県立高崎高等学校 二年 新井寿松 川越在人 石井佳大

概要・目的

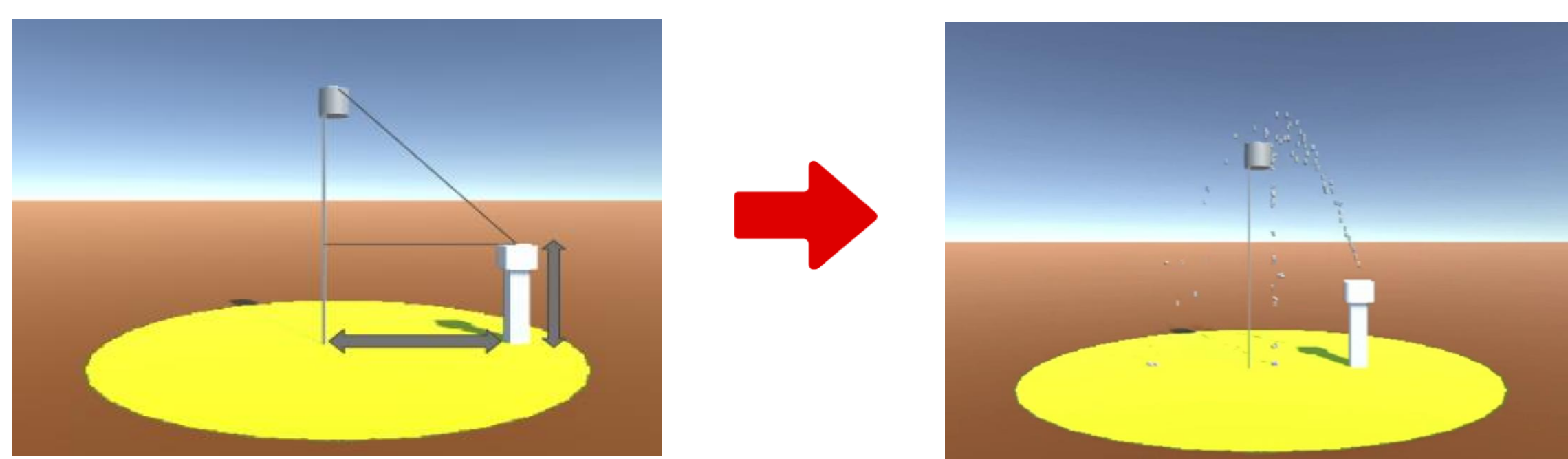
高崎高校と前橋高校では毎年「**定期戦**」と呼ばれる競技会を実施する。その中で**玉入れは最も配点の大きい競技の一つ**であり、**緻密な戦略とチームワークによる真剣勝負**が行われる。玉入れの先行研究は、調べてみてもほとんど無く、見つかったのは私たちの先輩が行った「玉入れの玉の散らばりの抑制」という研究のみだった。その研究では、玉を6つ同時に投げる際に玉の積み方によって散らばり具合が変化するかというものだったが、人に着目したものではなかった。そこで、**玉入れにおける人の効率的な動きについて研究することで**、定期戦を有利に進めることができるのではないかと考えた。玉入れではより多くの玉を入れるため、**個人戦略としてはどこからどのように投げるかが**、**チーム戦略としては一人ひとりがどのように動くか（玉を拾う人、投げる人の配分等）**が重要である。そこで本研究では、玉入れで玉が入りやすい場所を発見し、また、玉入れで効率良く動けるAIの作成を試みた。



手法と結果

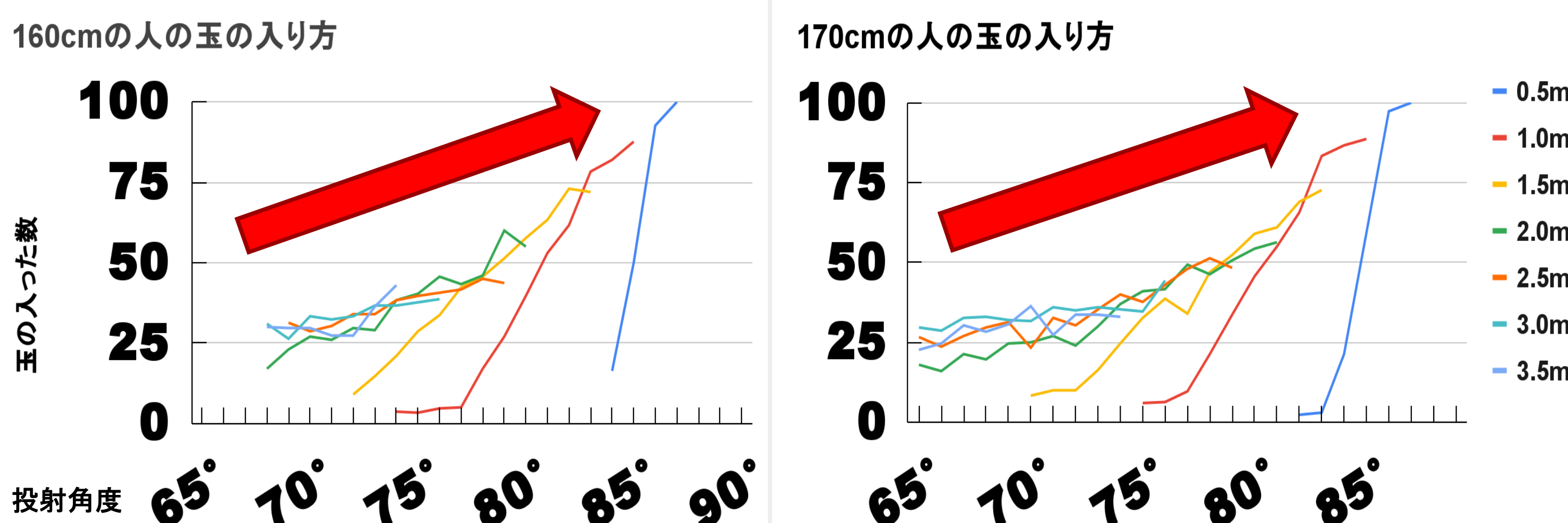
実験① 個人戦略としてどのように投げればよいか？

<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30cm上から、玉に初速度を与え斜方投射させる。その際、**初速度には誤差をつける**。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。



<条件>
・かごの大きさ 4.30m ・かごの直径 0.36m ・かごの深さ 0.38m ・コート直径 7.0m
・身長 160cmと170cm ・投射角度 65°~89°(1度ずつ) ・人とかごとの距離 0.5m~3.5m(0.5mずつ)
・初速度につける誤差 0.9倍~1.1倍 ・空気抵抗なし

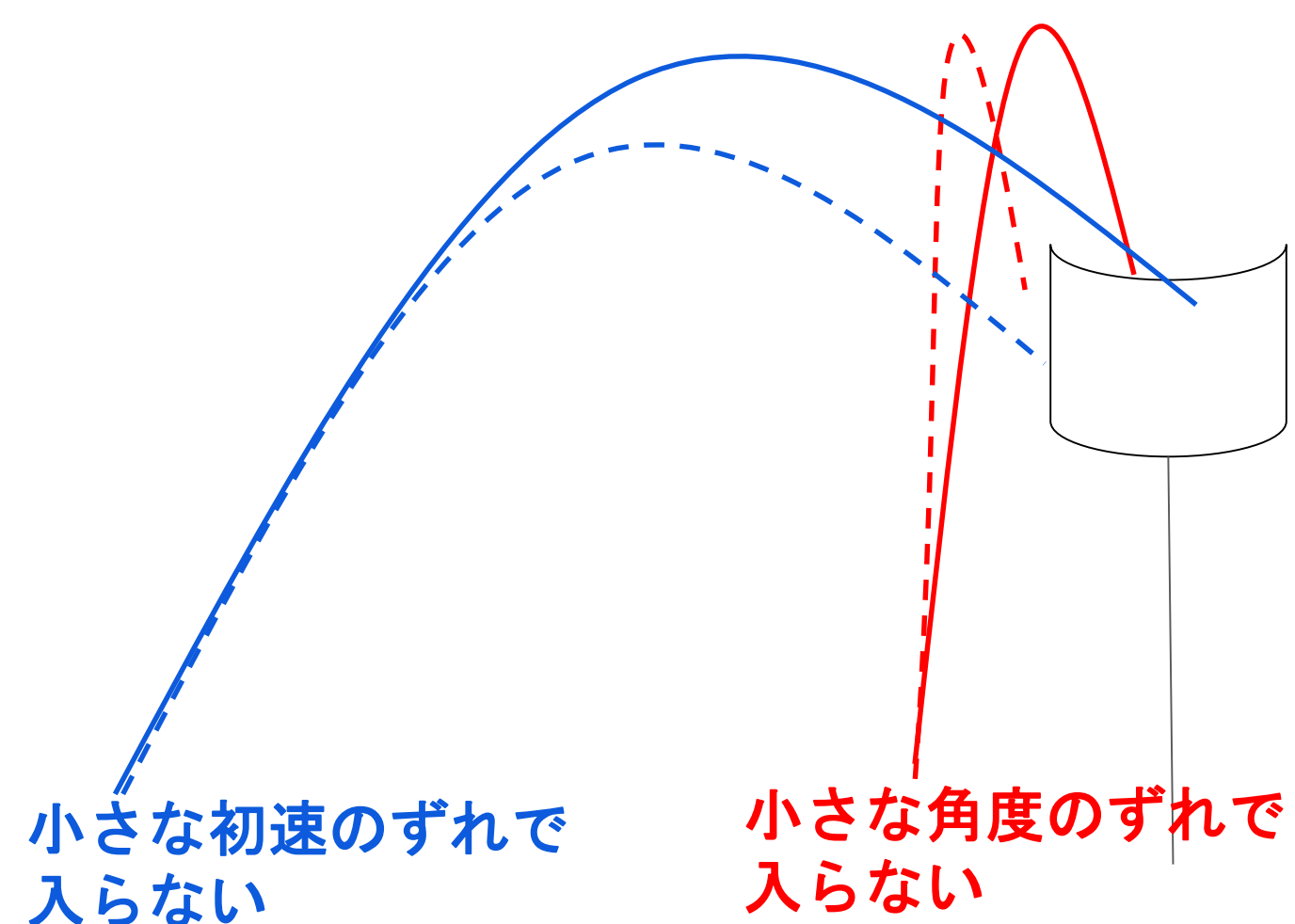
<結果>



- 身長に関係なく、かごからの距離が1.5m~2.0mから投げるとバランスよく入る
- 距離が遠すぎると角度及び身長に関係なく球が入りにくい。
- 距離が近くなると少し角度を変えるだけで入る玉の数が大きく変化するようになる

<考察>

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う、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



結果として、1.5~2.0mあたり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

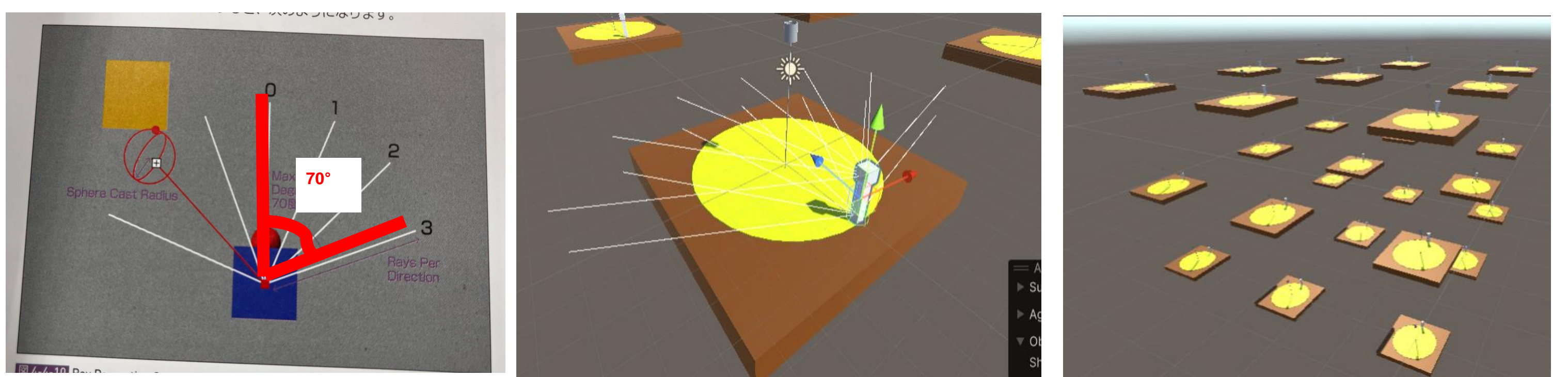
今後の展望

実験①では、1.5m~2.0mあたりから投げるのがよいと結論付けていたが、あくまでも、**シミュレーションによる考察のため、実測の実験による裏付け**もしていきたい。

実験②では、AIのパラメータ設定がうまくできておらず思い通りのAIが作成できなかったため、**学習ステップ数を増やす、報酬設定で玉を得たときの報酬を小さくする、または、その報酬をなしにする、もしくは、ML-Agentsの機能の一つであるカリキュラム学習**を使って、玉を拾うAIを作成してから玉を入れるAIを作成していくという2段階の学習など、AIの学習環境を変えて、もう一度AIの学習をしていきたい。

実験② チーム戦略としてどのように動けばよいか？

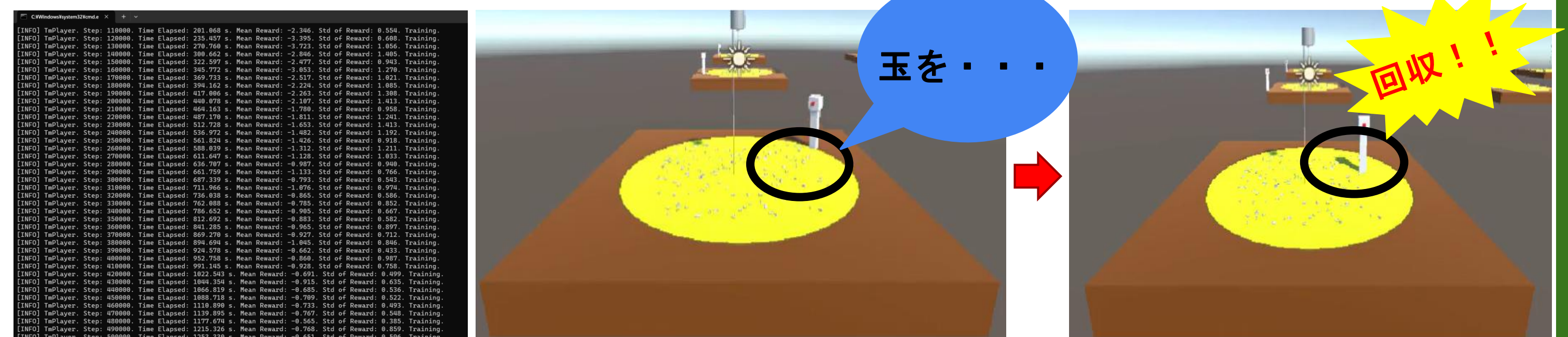
<方法> 物理エンジンのUnityを用いて実験①の玉入れのモデルを使い、玉入れの投げ方の最適化を行うAIを作成する。AIの作成にはUnityの機械学習ライブラリであるML-Agentsを使用した。



7本Rayを飛ばして、衝突したObjectを取得する。 足元と目線にRayを飛ばす。 モデルを27個置き、27倍の学習速度で行う。

<条件>
・AIの観察 RayCast Observation 7本x2(足元と目線) 片開70°(上図参照) Vector Observation(サイズ1) 0:持っている数(int)
・AIの行動 Continuous(サイズ6) 0,1:Playerのxz移動速 2:Playerのy軸回転速度 4,5:玉のxyz初速度 Discrete(サイズ1) 0:投げるかどうか(投げられないときは0固定)
・AIの報酬 玉が入ったら+1.0(1個につき) 玉を取得したら+0.01(1個につき) ステップ毎に-0.0005
・AIの決定タイミング 5ステップ毎 ・玉を拾う距離 20cm ・玉を持つ最大の個数 6個
・最大ステップ 500000ステップ ・空気抵抗は、速度の2乗に比例して近似

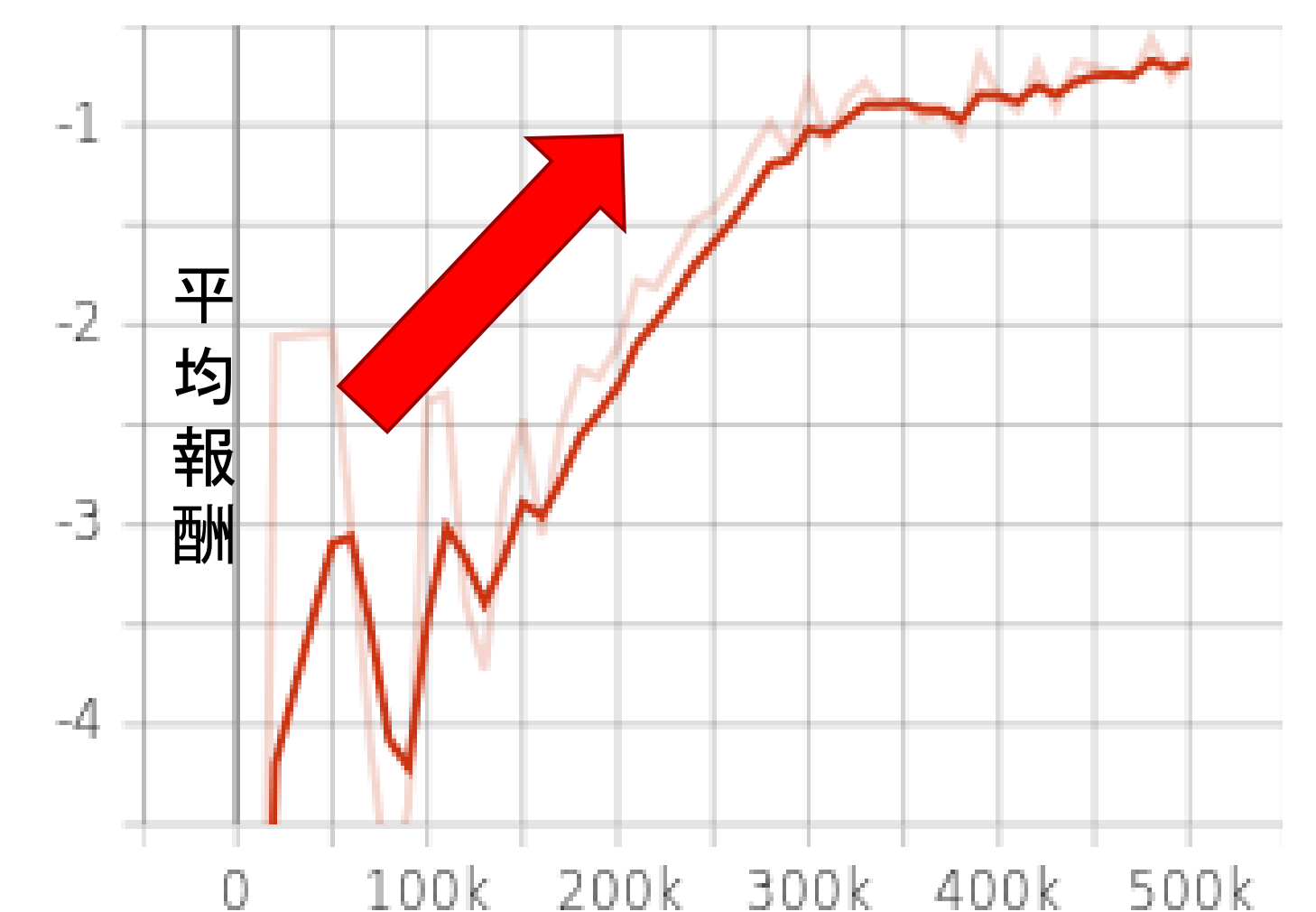
<結果>



AIの動作確認の画面

AIの動作確認の画面には、10000ステップごとに、平均報酬と標準偏差が表示される。

右の、ステップ数(時間)と平均報酬のグラフは、TensorBoardという機械学習ワークフローの際に必要な測定値と視覚化を提供するツールを用いて作成されたものである。



<考察>

グラフの通り、**時間がたつほど平均報酬が増えているため、AIの作成は成功したように見える**。しかし、学習したモデルを確認してみると、実際は、**ボールを拾う報酬のみ最適化が行われた、つまり、ボールを拾うだけで、かごに向かって球を投げないAIが作成されていた**。このようなAIが作成された要因は、AIの報酬などのパラメータ設定がうまくできなかったからであると考えられる。

動作環境・参考文献

動作環境

- ・ Microsoft Windows 11 Home ・ CPU 12th Gen Intel(R) Core(TM) i5-12500H
- ・ GPU NVIDIA GeForce RTX 3050 Laptop ・ Unity 2022.3.4f1 ・ ML-Agents Release 19
- ・ Python 3.8.18

参考文献

- 1)群馬県立高崎高等学校 飯島拓真,石井雅人,小山田匠吾 「玉入れの玉の散らばりの抑制」
- 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,ポーンデジタル
- 3)<https://www.eee.kagoshima-u.ac.jp/~watanabe-lab/Phys2019/%E7%89%A9%E7%90%86%E5%AD%A6%E5%9F%BA%E7%A4%8EAI-2019-3-R2.pdf> (鹿児島大学 工学部 電気電子工学科 物理学基礎AI(第3回)一運動の法則(2)一)