

強化学習を利用したゲームの設計

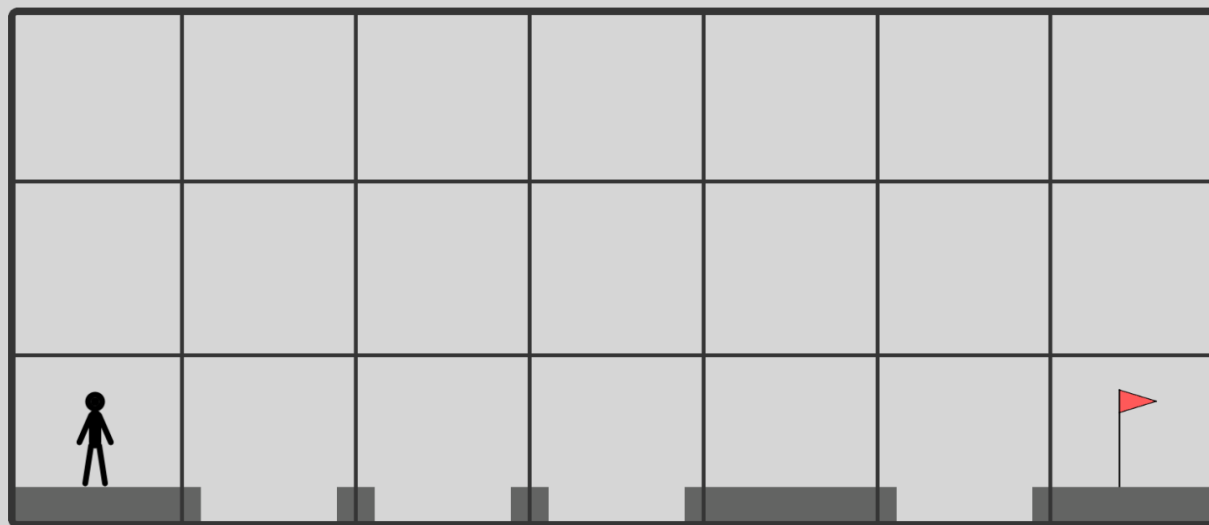


研究の概要



研究内容

- プレイヤーがステージを組み立てて、ゲーム内AIがステージをプレイするゲームを開発する
- AIの学習には強化学習を用いる



研究目標

- ゲームの完成
 - ・ ゲーム内AIとプレイヤーがステージを設計する機能の実装でもって完成
- 効率の良い学習の方法を決める
 - ・ 効率よく学習するための報酬の与え方やパラメータの設定を考える
- 最終的なAIの動きを観察する

研究の動機

ゲーム制作の中で、ステージ設計は難しく大変な作業



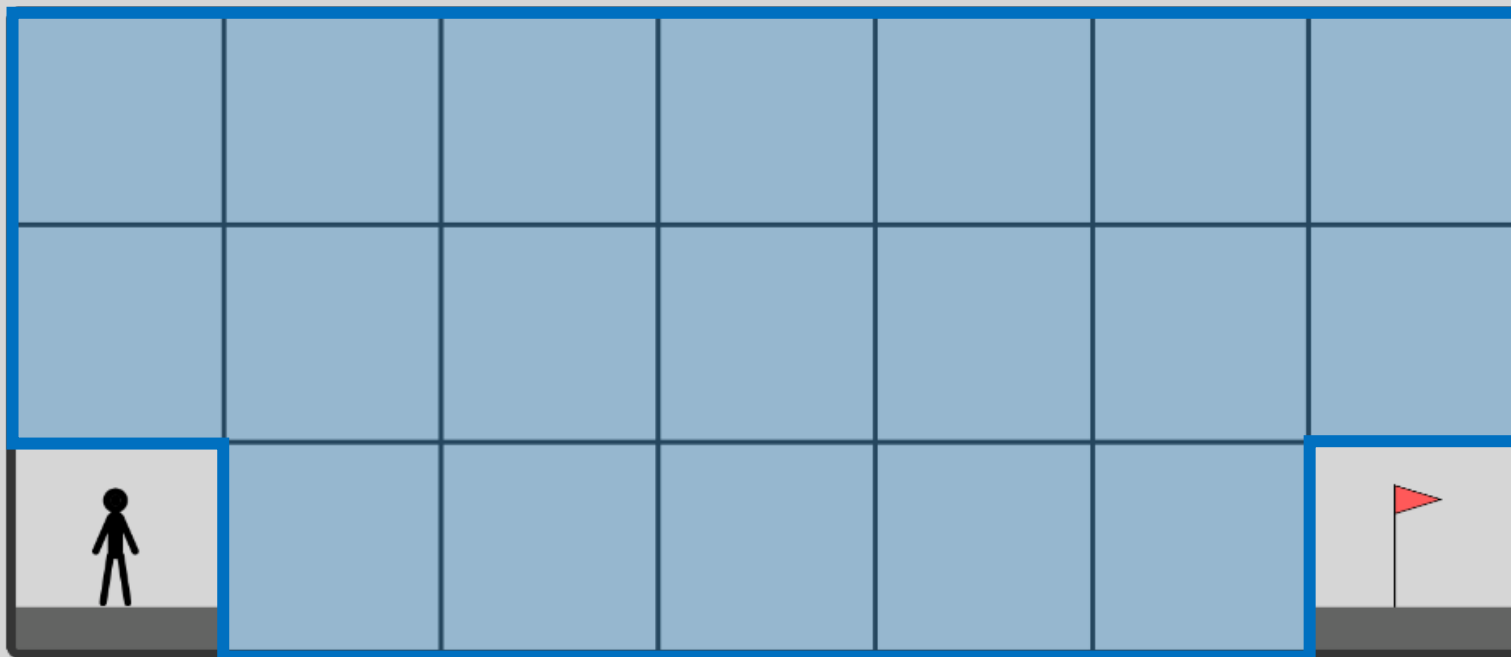
プレイヤーにステージ設計をさせることで負担を軽減

ゲームの詳細



ゲームシステム①

- プレイヤーは決められたマス目の中にステージ部品を設置する
 - 下図は初期状態
 - 青で囲われた部分が設置可能なマス目
- 完成したら自分でプレイし、クリアできたらAIにプレイさせる



ゲームシステム②

- AIプレイ時、下記の条件でポイントが加算されていく
 - ・ 経過時間
 - ・ ゲームオーバー
 - ・ 制限時間内にクリアできなかった時
- つまり、AIにとって難しいステージを作ったら勝ち

開発環境

- ゲームエンジン：Unity
 - ・ Unityでの強化学習の実装はML_Agentsを使う
- 使用言語：C#, YAML

学習の詳細



強化学習の手法

- PPO (Proximal Policy Optimization)
 - 方策勾配法を用いる手法の一つ
 - ML-Agentsではデフォルトのアルゴリズム

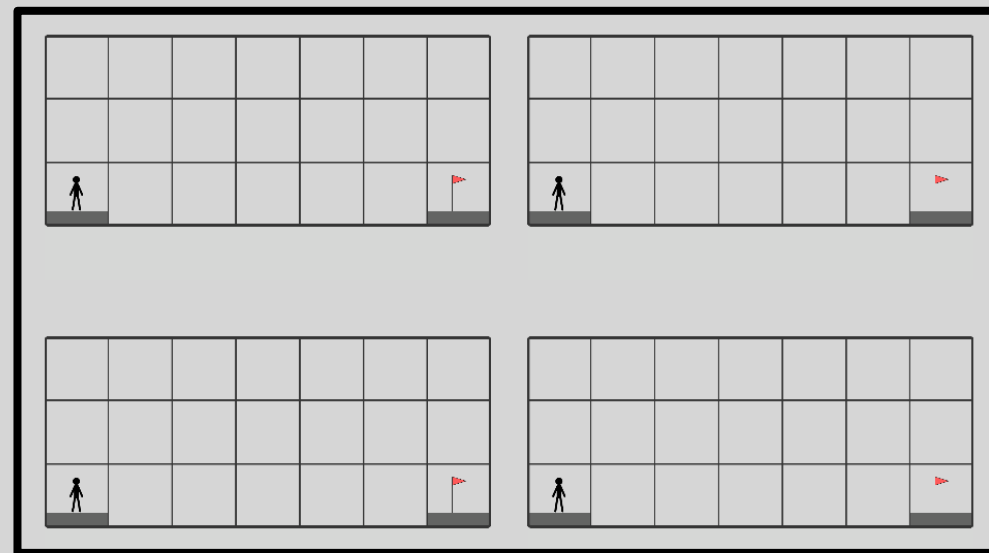
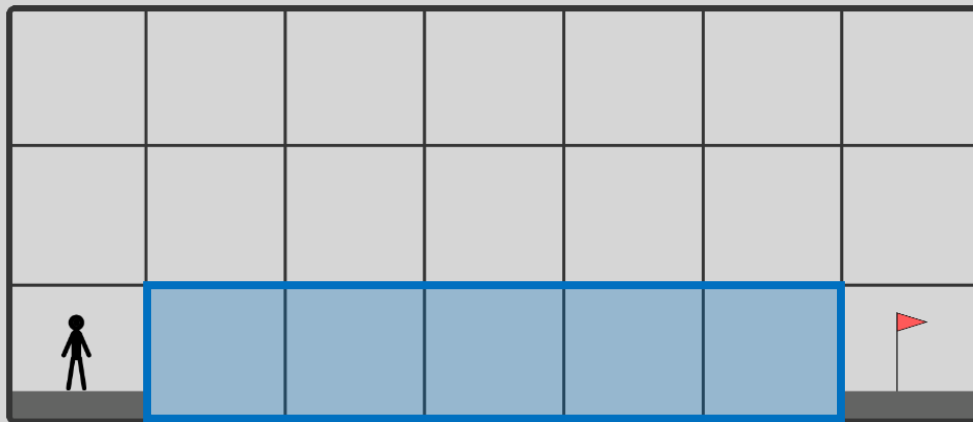
$$L^{CLIP}(\theta) = \hat{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right]$$

環境

- 7×3のマス目の中にランダムでステージ部品を設置する
 - ・ 今回は一番下のマス目にのみステージ部品を設置している
 - ・ Ground_1の発生確率は60%、Ground_2の発生確率は40%
 - ・ 学習効率化の為、環境は4つで行う

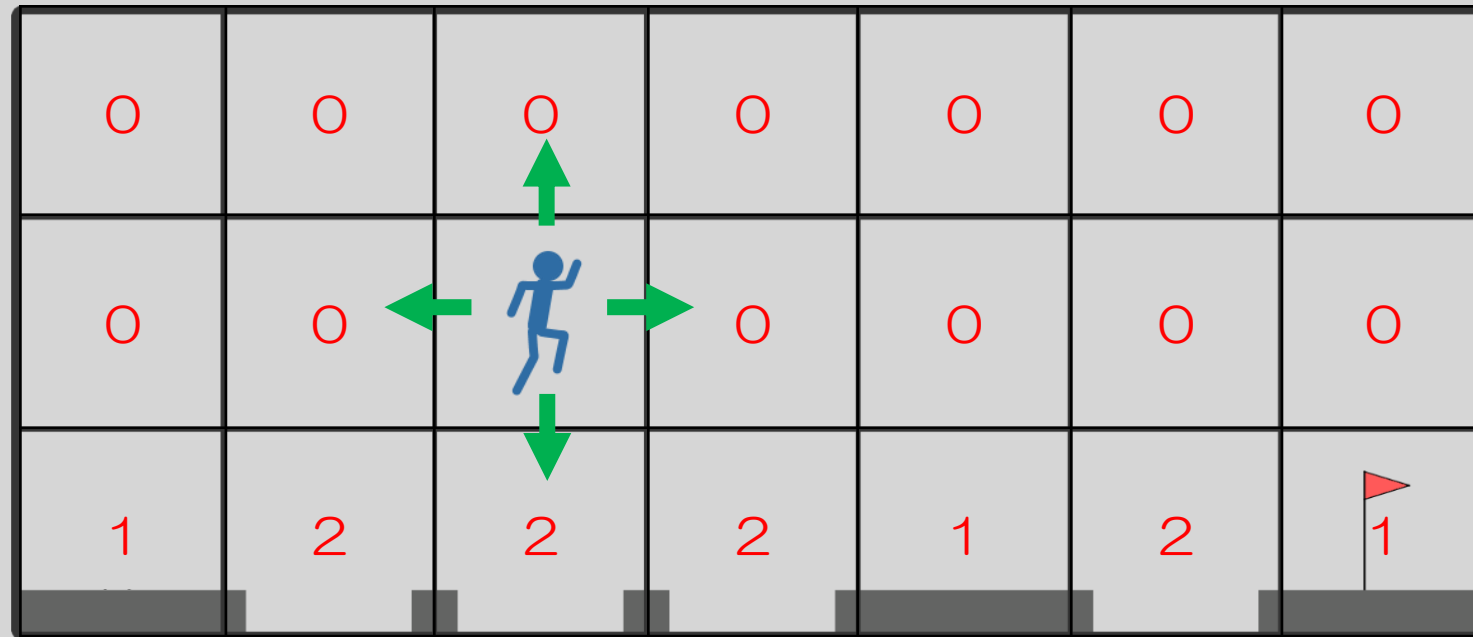
部品名：Ground_1

部品名：Ground_2



状態

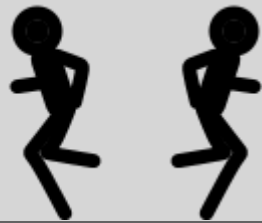
- AIがいるマスにあるステージ部品 (`partsID`)
- AIにかかっている速度 (`transform.velocity`)
- AIの位置 (`transform.position.x`, `transform.position.y`)



行動

- 左右の横移動とジャンプのみ

左右横移動

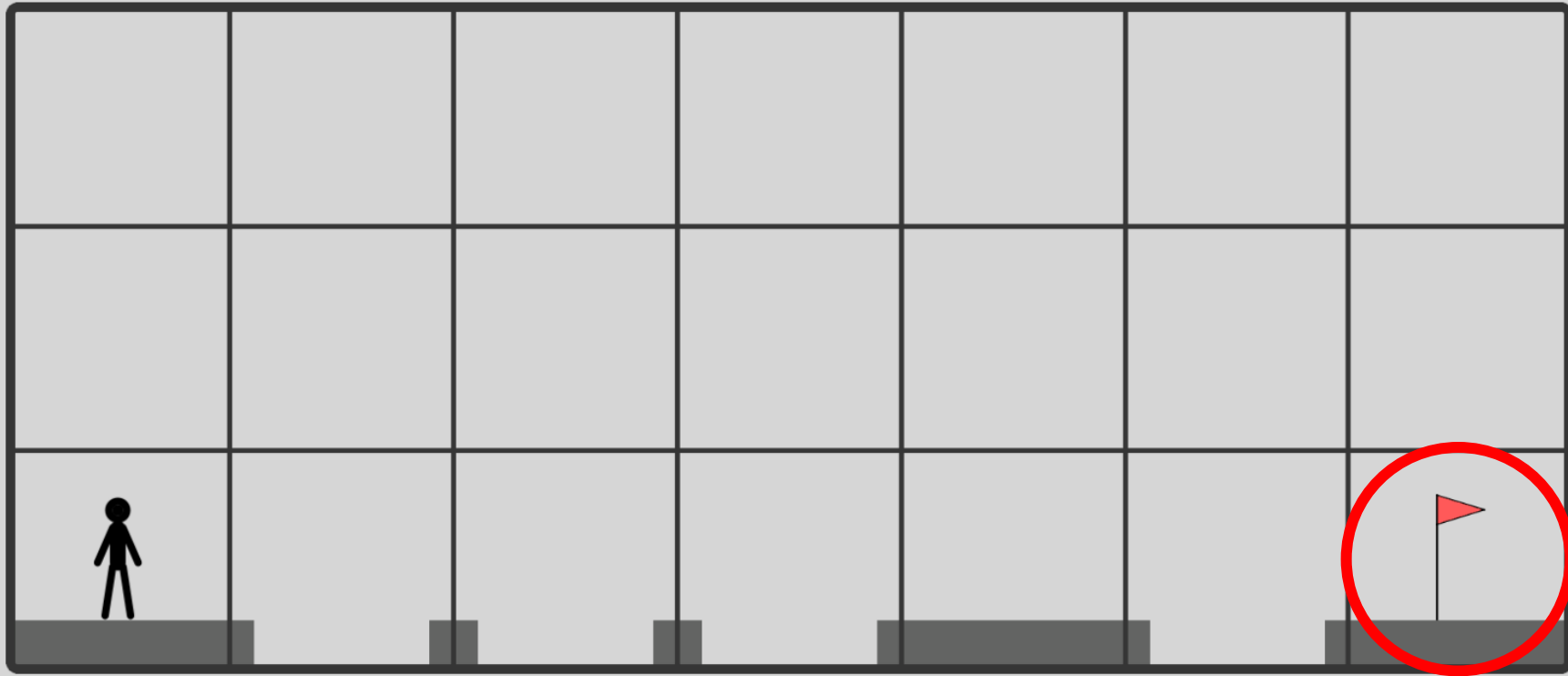


ジャンプ



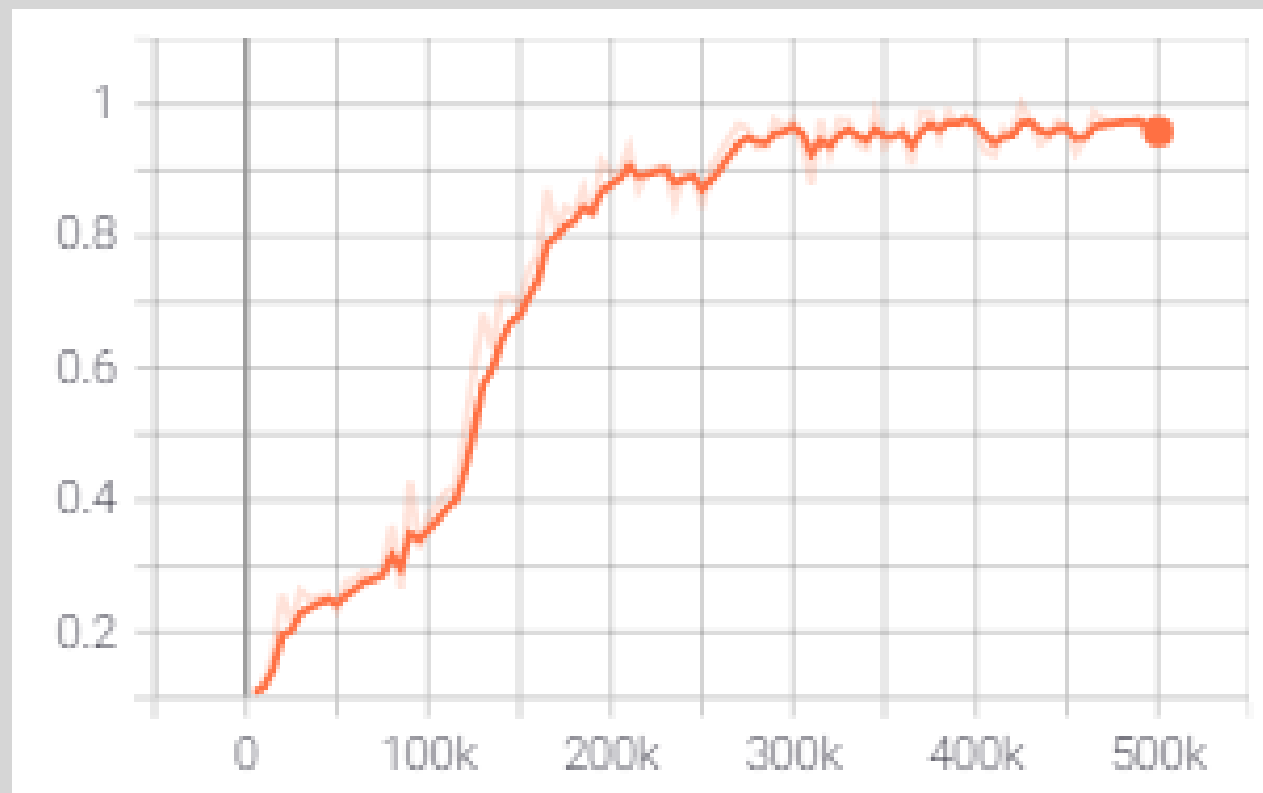
報酬

- ステージ右下の旗に触れれば報酬を与える (+1.0)

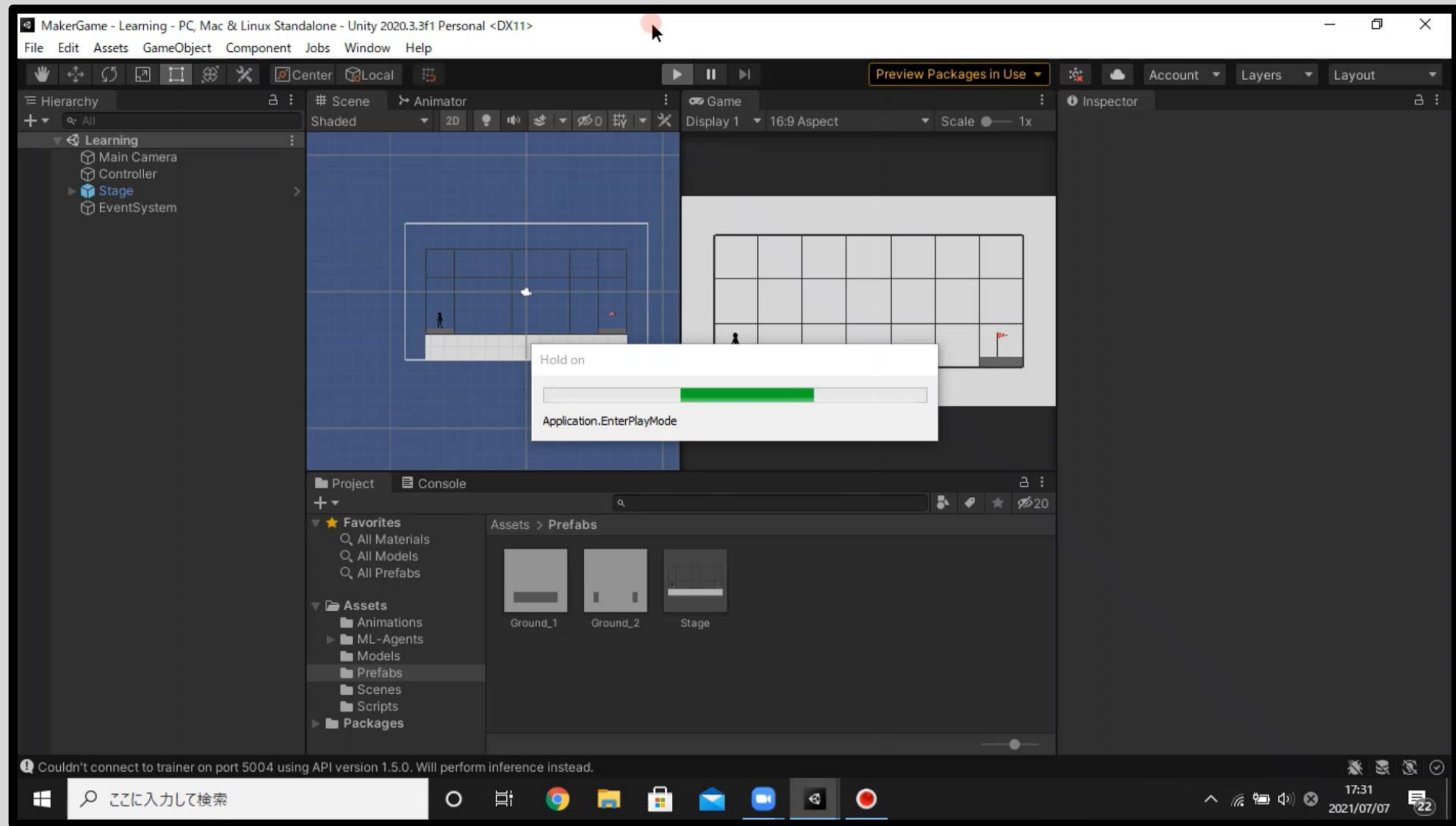


学習の結果

- 報酬の平均は0.9あたりで飽和し、うまく学習できた



実際のAIの動き



日程計画



これからの予定

- 設置するステージ部品の範囲を広げる
- より多くのステージ部品を追加する
- ステージ設計の機能を実装する
- AIプレイの機能を実装する

日程

- 7～8月（21年）
 - ・ 設置するステージ部品の範囲を広げる
 - ・ より多くのステージ部品を追加する
- 9～10月（21年）
 - ・ ステージ設計の機能を実装する
- 11～12月（21年）
 - ・ AIプレイの機能を実装する
 - ・ ゲーム完成
- 1～3月（22年）
 - ・ 論文作成

ご清聴ありがとうございました